



TUGAS AKHIR - TF145565

**RANCANG BANGUN SISTEM *MONITORING*
JARAK JAUH PADA *HYDROSTATIC TRAINING*
SYSTEM PLANT SKALA LABORATORIUM**

ANDIAH AGUSTININGRUM

NRP. 1051 14 0000 0052

Dosen Pembimbing

Ir. Ya'umar, MT

NIP. 19540406 198103 1 003

Dwi Oktavianto Wahyu Nugroho, ST, MT

NPP. 19832017 11054

Program Studi DIII Teknologi Instrumentasi

Departemen Teknik Instrumentasi

Fakultas Vokasi

Institut Teknologi Sepuluh Nopember

Surabaya

2018



TUGAS AKHIR - TF145565

**RANCANG BANGUN SISTEM *MONITORING*
JARAK JAUH PADA *HYDROSTATIC TRAINING*
SYSTEM PLANT SKALA LABORATORIUM**

ANDIAH AGUSTININGRUM

NRP. 1051 14 0000 0052

Dosen Pembimbing

Ir. Ya'umar, MT

NIP. 19540406 198103 1 003

Dwi Oktavianto Wahyu Nugroho, ST, MT

NPP. 19832017 11054

Program Studi DIII Teknologi Instrumentasi

Departemen Teknik Instrumentasi

Fakultas Vokasi

Institut Teknologi Sepuluh Nopember

Surabaya

2018



FINAL PROJECT - TF145565

**DESIGN OF MOBILE MONITORING SYSTEM
FOR HYDROSTATIC TRAINING SYSTEM PLANT
LABORATORY SCALE**

ANDIAH AGUSTININGRUM

NRP. 1051 14 0000 0052

Supervisor

Ir. Ya'umar, MT

NIP. 19540406 198103 1 003

Dwi Oktavianto Wahyu Nugroho, ST, MT

NPP. 19832017 11054

Study Program DIII Instrumentation Technology

Department of Instrumentation Engineering

Faculty of Vocation

Institut Teknologi Sepuluh Nopember

Surabaya

2018

**RANCANG BANGUN SISTEM *MONITORING* JARAK JAUH
PADA *HYDROSTATIK TRAINING SYSTEM PLANT*
SKALA LABORATORIUM**

TUGAS AKHIR

Oleh :

Andiah Agustiningrum

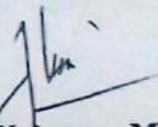
NRP. 10511400000052

Surabaya, 1 Agustus 2018

Menyetujui,

Dosen Pembimbing I

Dosen Pembimbing II


Ir. Ya'umar, MT

NIP. 19540406 199802 2 001


Dwi Oktavianto W.N., ST, MT

NPP. 19832017 11054

Mengetahui,

Kepala Departemen Teknik Instrumentasi FV-ITS




Dr. Ir. Purwadi Agus Darwito, M.Sc

NIP. 19620822 198803 1 001

**RANCANG BANGUN SISTEM MONITORING JARAK JAUH
PADA HYDROSTATIK TRAINING SYSTEM PLANT
SKALA LABORATORIUM**

TUGAS AKHIR

**Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Ahli Madya
Pada**

**Program Studi DIII Teknologi Instrumentasi
Departemen Teknik Instrumentasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember**

Oleh :

Andiah Agustiningrum

NRP. 10511400000052

Disetujui Oleh Tim Penguji Tugas Akhir :

1. Ir. Ya'umar, MT

..... (Pembimbing I)

2. Dwi Oktavianto W. N., ST, MT

..... (Pembimbing II)

3. Ir. Tutug Dhanardono, MT

..... (Penguji)

**SURABAYA
AGUSTUS 2018**

RANCANG BANGUN SISTEM *MONITORING* JARAK JAUH PADA *HYDROSTATIC TRAINING SYSTEM PLANT* SKALA LABORATORIUM

Nama : Andiah Agustiningrum
NRP : 10511400000052
Program Studi : DIII Teknologi Instrumentasi
Pembimbing I : Ir. Ya'umar, MT
Pembimbing II : Dwi Oktavianto Wahyu Nugroho, ST, MT

ABSTRAK

Telah dibuat sistem *monitoring* jarak jauh pada *hydrostatic training system plant* skala laboratorium untuk *variable level* dan *flow* menggunakan modul Wifi ESP8266 sebagai *transmitter* data dari sensor ke *database MySQL web server* yang ditampilkan pada aplikasi android.

Dari pengambilan data dengan 3 jarak yang berbeda didapatkan hasil pembacaan pada jarak pancar ESP8266 4 m dari *plant*, error maksimum *level* 1 0.41 cm dan *level* 2 1.15 cm dengan *delay* pengiriman data 3 detik, error maksimum *flow* 1 0.06 l/s dan *flow* 2 0.09 l/s dengan *delay* pengiriman data 3 detik. 8 m dari *plant*, error maksimum *level* 1 1.36 cm dan *level* 2 1.49 cm dengan *delay* pengiriman data 5 detik, *flow* 1 0.57 l/s dan *flow* 2 0.07 l/s dengan *delay* pengiriman data 4 detik. 10 m dari *plant*, error maksimum *level* 1 1.66 cm dan *level* 2 2.47 cm dengan *delay* pengiriman data 9 detik, *flow* 1 0.65 l/s dan *flow* 2 0.8 l/s dengan *delay* pengiriman data 9 detik.

Sehingga dapat disimpulkan bahwa jarak pengiriman data ESP8266 dari *plant* berbanding lurus dengan error maksimum data dan *delay* pengiriman data.

Kata Kunci: *monitoring*, jarak jauh, *flow*, *level*, *database*, ESP8266, android, error, *delay*

**DESIGN OF MOBILE MONITORING SYSTEM
FOR HYDROSTATIC TRAINING SYSTEM PLANT
LABORATORY SCALE**

Name : Andiah Agustiningrum
NRP : 10511400000052
Study Program : *DIH Instrumentation Technology*
Supervisor I : Ir. Ya'umar, MT
Supervisor II : Dwi Oktavianto Wahyu Nugroho, ST, MT

ABSTRACT

A mobile monitoring system for level and flow variable for laboratory scale of hydrostatic training system plant was created. ESP8266 as data transmitter module from sensor to MySQL web server database and the database will be displayed on android application.

Based on data experiment result with 3 different ESP8266 transmit distances at 4 m from plant, maximum error for level 1 is 0.41 cm and level 2 is 1.15 cm with 3 second data transmission delay, then maximum error for flow 1 is 0.06 l/s and flow 2 is 0.09 l/s with 3 seconds data transmission delay. At 8 m from plant, error maximum level 1 is 1.36 cm and level is 2 is 1.49 cm with 5 seconds data transmission delay, then maximum error for flow 1 is 0.57 l/s and flow 2 is 0.07 l/s with 4 seconds data transmission delay. At 10 m from plant error maximum level 1 is 1.66 cm and level 2 is 2.47 cm with 9 seconds second data transmission delay, then maximum error for flow 1 is 0.65 l/s and flow 2 is 0.8 l/s with 9 seconds data transmission delay.

So it can be concluded that the ESP8266 transmit distance from plant same as maximum error value and data transmission delay.

Keywords: monitoring, mobile, flow, level, database, ESP8266, android, error, delay

KATA PENGANTAR

Alhamdulillah, puji syukur kita panjatkan kehadiran Allah SWT karena limpahan rahmat dan karunia-Nya kepada penulis sehingga dapat menyelesaikan Tugas Akhir dengan judul :

RANCANG BANGUN SISTEM *MONITORING* JARAK JAUH PADA *HYDROSTATIC TRAINING SYSTEM PLANT* SKALA LABORATORIUM

Tugas Akhir dibuat sebagai salah satu syarat mendapatkan gelar ahli madya program studi DIII Teknologi Instrumentasi, Departemen Teknik Instrumentasi, Fakultas Vokasi Institut Teknologi Sepuluh Nopember.

Selama pembuatan Tugas Akhir, penulis tidak lepas dari bantuan dan dukungan berbagai pihak. Untuk itu, pada kesempatan kali ini penulis ingin mengucapkan terima kasih kepada :

1. Keluarga dan sahabat penulis yang memberikan dukungan sehingga dapat menyelesaikan studi DIII.
2. Bapak Dr. Ir. Purwadi Agus Darwito, M.Sc selaku Ketua Departemen Teknik Instrumentasi ITS.
3. Bapak Ir. Ya'umar, MT dan bapak Dwi Oktavianto Wahyu Nugroho, ST, MT selaku dosen pembimbing yang telah memberikan bimbingan, kritik dan saran selama pembuatan tugas akhir ini.
4. Ibu Dr.-Ing. Doty Dewi Risanti, ST, MT selaku dosen wali yang telah memberikan bimbingan, kritik dan saran selama kuliah.
5. Kelompok Hydrostatic yang tetap teguh ingin lulus #118 walaupun banyak rintangan dan tantangan.

6. Teman-teman DIII Teknik Instrumentasi dan Teknik Fisika Angkatan 2014 dan Angkatan 2015 yang telah membagi ilmu, pengalaman, membantu dan memberikan dukungan.
7. Serta pihak-pihak lain yang turut membantu dalam pembuatan tugas akhir ini yang tidak dapat disebutkan satu per satu.

Dalam pembuatan tugas akhir ini masih banyak terdapat kekurangan dan jauh dari sempurna. Oleh karena itu, penulis memohon kritik dan saran dari pembaca agar laporan ini mendekati sempurna. Akhirul kalam semoga tugas akhir ini dapat bermanfaat bagi saya dan para pembaca di lingkungan Teknik Instrumentasi FV-ITS.

Surabaya, 1 Agustus 2018

Penulis

Andiah Agustiningrum
NRP. 10511400000052

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN	iii
ABSTRAK	v
ABSTRACT	vii
KATA PENGANTAR	ix
DAFTAR ISI	xi
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xvii
BAB I PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Perumusan Masalah	1
1.3 Batasan Masalah	1
1.4 Tujuan	2
1.5 Sistematika Pelaporan	2
BAB II DASAR TEORI	
2.1 Fluida	3
2.2 Tekanan Hidrostatik	7
2.3 Sistem Monitoring	7
2.4 Sensor	9
2.5 Mikrokontroler AVR Atmega 1284P	12
2.6 Modul WiFi ESP8266	14
2.7 Database	15
2.8 Android	16
2.9 Android Studio	17
2.10 Karakteristik Statik	18
2.11 Teori Ketidakpastian	21
BAB III METODOLOGI	
3.1 Alat dan Bahan	25
3.2 Prosedur Percobaan	27

BAB IV HASIL DAN PEMBAHASAN

4.1 Hasil Pembuatan Alat	33
4.2 Data Hasil Pengujian Alat	38
4.3 Pembahasan.....	65

BAB V PENUTUP

5.1 Kesimpulan	67
5.2 Saran.....	68

DAFTAR PUSTAKA

69

Lampiran A

Lampiran B

Lampiran C

Lampiran D

Lampiran E

Lampiran F

Lampiran G

DAFTAR GAMBAR

Gambar 2.1 Hukum Kontinuitas Bernoulli	3
Gambar 2.2 Perbedaan Aliran Fluida	5
Gambar 2.3 Diagram Moody	6
Gambar 2.4 Diagram Blok Sistem <i>Monitoring</i>	8
Gambar 2.5 Konfigurasi Pin Sensor Ultrasonik HC-SR04.....	9
Gambar 2.6 Cara Kerja Sensor Ultrasonik HC-SR04.....	10
Gambar 2.7 Konfigurasi Kabel <i>Water Flow Sensor</i>	11
Gambar 2.8 Cara Kerja <i>Water Flow Sensor</i>	12
Gambar 2.9 Konfigurasi Pin AVR Atmega1284P	14
Gambar 2.10 Konfigurasi Pin ESP8266	15
Gambar 2.11 Arsitektur dan Komunikasi <i>Client-Server</i>	16
Gambar 2.12 Android Studio	18
Gambar 2.13 Karakteristik Linier <i>Output</i> Instrumen	19
Gambar 2.14 Grafik Non-Linieritas.....	20
Gambar 2.15 Hubungan Akurasi dan Presisi.....	21
Gambar 3.1 Sensor Ultrasonik HC-SR04.....	25
Gambar 3.2 <i>Water Flow Sensor</i>	25
Gambar 3.3 AVR Atmega1284P.....	26
Gambar 3.4 ESP8266.....	26
Gambar 3.5 Android Studio.....	26
Gambar 3.6 Diagram Alir Pembuatan Tugas Akhir	27
Gambar 3.7 Diagram Blok Sistem <i>Monitoring</i> Jarak Jauh.....	28
Gambar 3.8 <i>PFD</i> Sistem <i>Monitoring</i> Jarak Jauh Pada <i>Hydrostatic Training System</i> Skala Laboratorium..	29
Gambar 3.9 <i>P&ID</i> Sistem <i>Monitoring</i> Jarak Jauh Pada <i>Hydrostatic Training System</i> Skala Laboratorium .	30
Gambar 3.10 Alat <i>Hydrostatic Training System</i> Skala Laboratorium	31
Gambar 4.1 ESP8266 Pada Board.....	33

Gambar 4.2 Diagram Alir Serial ESP8266	34
Gambar 4.3 Pembuatan <i>Database MySQL</i>	36
Gambar 4.4 Tampilan Aplikasi Android	37
Gambar 4.5 Grafik Perbandingan Alat Ukur Standar dan Uji sensor Ultrasonik HC-SR04	40
Gambar 4.6 Grafik Perbandingan Pembacaan <i>Level</i> Sistem A Di LCD dan Android Pada Jarak Pancar ESP8266 4m dari <i>Plant</i>	47
Gambar 4.7 Grafik Perbandingan Pembacaan <i>Level</i> Sistem B Di LCD dan Android Pada Jarak Pancar ESP8266 4m dari <i>Plant</i>	48
Gambar 4.8 Grafik Perbandingan Pembacaan <i>Flow</i> Sistem A Di LCD dan Android Pada Jarak Pancar ESP8266 4m dari <i>Plant</i>	49
Gambar 4.9 Grafik Perbandingan Pembacaan <i>Flow</i> Sistem B Di LCD dan Android Pada Jarak Pancar ESP8266 4m dari <i>Plant</i>	49
Gambar 4.10 Grafik Perbandingan Pembacaan <i>Level</i> Sistem A Di LCD dan Android Pada Jarak Pancar ESP8266 8m dari <i>Plant</i>	55
Gambar 4.11 Grafik Perbandingan Pembacaan <i>Level</i> Sistem B Di LCD dan Android Pada Jarak Pancar ESP8266 8m dari <i>Plant</i>	55
Gambar 4.12 Grafik Perbandingan Pembacaan <i>Flow</i> Sistem A Di LCD dan Android Pada Jarak Pancar ESP8266 8m dari <i>Plant</i>	57
Gambar 4.13 Grafik Perbandingan Pembacaan <i>Flow</i> Sistem B Di LCD dan Android Pada Jarak Pancar ESP8266 8m dari <i>Plant</i>	57

Gambar 4.14 Grafik Perbandingan Pembacaan <i>Level</i> Sistem A Di LCD dan Android Pada Jarak Pancar ESP8266 10m dari <i>Plant</i>	61
Gambar 4.15 Grafik Perbandingan Pembacaan <i>Level</i> Sistem A Di LCD dan Android Pada Jarak Pancar ESP8266 10m dari <i>Plant</i>	62
Gambar 4.16 Grafik Perbandingan Pembacaan <i>Flow</i> Sistem A Di LCD dan Android Pada Jarak Pancar ESP8266 10m dari <i>Plant</i>	63
Gambar 4.17 Grafik Perbandingan Pembacaan <i>Flow</i> Sistem B Di LCD dan Android Pada Jarak Pancar ESP8266 10m dari <i>Plant</i>	64

Halaman ini memang dikosongkan

DAFTAR TABEL

Tabel 2.1 Spesifikasi Sensor Ultrasonik	9
Tabel 2.2 Spesifikasi Water Flow Sensor	11
Tabel 2.3 Spesifikasi AVR Atmega 1284P	13
Tabel 2.4 Spesifikasi Modul WiFi ESP8266	14
Tabel 4.1 Hasil Pengujian Naik Sensor Ultrasonik HC-SR04 ..	38
Tabel 4.2 Hasil Pengujian Turun Sensor Ultrasonik HC-SR04 .	39
Tabel 4.3 Data Pembacaan Sistem A Setpoint 25cm Di Android Pada Jarak Pancar ESP8266 4m dari <i>Plant</i>	42
Tabel 4.4 Data Pembacaan Sistem B Setpoint 25cm Di Android Pada Jarak Pancar ESP8266 4m dari <i>Plant</i>	43
Tabel 4.5 Data Pembacaan Sistem A Setpoint 25cm Di LCD Pada Jarak Pancar ESP8266 4m dari <i>Plant</i>	44
Tabel 4.6 Data Pembacaan Sistem B Setpoint 25cm Di LCD Pada Jarak Pancar ESP8266 4m dari <i>Plant</i>	45
Tabel 4.7 Perbandingan Data Pembacaan <i>Level</i> Setpoint 25 cm LCD dan Android Pada Jarak Pancar ESP8266 4m dari <i>Plant</i>	46
Tabel 4.8 Perbandingan Data Pembacaan <i>Flow</i> Setpoint 25 cm LCD dan Android Pada Jarak Pancar ESP8266 4m dari <i>Plant</i>	48
Tabel 4.9 Data Pembacaan Sistem A Setpoint 25cm Di Android Pada Jarak Pancar ESP8266 8m dari <i>Plant</i> .	50
Tabel 4.10 Data Pembacaan Sistem B Setpoint 25cm Di Android Pada Jarak Pancar ESP8266 8m dari <i>Plant</i> .	51
Tabel 4.11 Data Pembacaan Sistem A Setpoint 25 cm Di LCD Pada Jarak Pancar ESP8266 8m dari <i>Plant</i> . .	52
Tabel 4.12 Data Pembacaan Sistem B Setpoint 25 cm Di LCD Pada Jarak Pancar ESP8266 8m dari <i>Plant</i> . .	53

Tabel 4.13 Perbandingan Data Pembacaan <i>Level</i> Setpoint 25cm LCD dan Android Pada Jarak Pancar ESP8266 8m dari <i>Plant</i>	54
Tabel 4.14 Perbandingan Data Pembacaan <i>Flow</i> Setpoint 25cm LCD dan Android Pada Jarak Pancar ESP8266 8m dari <i>Plant</i>	56
Tabel 4.15 Data Pembacaan Sistem A Setpoint 25cm Di Android Pada Jarak Pancar ESP8266 10m dari <i>Plant</i>	58
Tabel 4.17 Data Pembacaan Sistem A Setpoint 25cm Di LCD Pada Jarak Pancar ESP8266 10m dari <i>Plant</i>	59
Tabel 4.16 Data Pembacaan Sistem B Setpoint 25cm Di Android Pada Jarak Pancar ESP8266 10m dari <i>Plant</i>	59
Tabel 4.18 Data Pembacaan Sistem B Setpoint 25cm Di LCD Pada Jarak Pancar ESP8266 10m dari <i>Plant</i>	60
Tabel 4.19 Perbandingan Data Pembacaan <i>Level</i> Setpoint 25cm LCD dan Android Pada Jarak Pancar ESP8266 10m dari <i>Plant</i>	61
Tabel 4.20 Perbandingan Data Pembacaan <i>Flow</i> Setpoint 25cm LCD dan Android Pada Jarak Pancar ESP8266 10m dari <i>Plant</i>	63

Bab I

Pendahuluan

1.1 Latar Belakang

Tekanan hidrostatik banyak dimanfaatkan untuk kegiatan produksi di industri. Oleh karena itu dibuat sebuah Rancang Bangun *Hydrostatic Training System Plant* Skala Laboratorium untuk membantu mahasiswa memahami dan mengaplikasikan ilmu tekanan hidrostatik.

Pada Rancang Bangun *Hydrostatic Training System Plant* Skala Laboratorium ini memiliki dua variabel input yaitu *level* dan *flow* dengan menggunakan dua sistem pengendalian yaitu dan sistem pengendalian umpan balik (*feed back*) pada sistem A dan sistem B. Perbedaan pada kedua sistem ini adalah penggunaan *MOV (motor operated valve)* sebagai aktuator sistem A dan pompa air sebagai aktuator sistem B. Selain itu juga dirancang sistem *safety* yang menggunakan *solenoid valve (sv)* sebagai katup pembuangan (*drain valve*). Perbedaan nilai *level* fluida pada tangki A dan B digunakan untuk mencari nilai tekanan hidrostatiknya.

Untuk memastikan proses sesuai harapan dan mengetahui perkembangan pencapaian target digunakan sistem *monitoring* jarak jauh dalam bentuk visual secara *mobile* pada android sebagai latar belakang dibuatnya “Rancang Bangun Sistem *Monitoring* Jarak Jauh Pada *Hydrostatic Training System Plant* Skala Laboratorium”.

1.2 Perumusan Masalah

Berdasarkan latar belakang yang dijelaskan diatas, maka rumusan masalah dalam tugas akhir ini adalah bagaimana cara merancang sistem *monitoring* jarak jauh pada *hydrostatic training system plant* skala laboratorium?

1.3 Batasan Masalah

Untuk memfokuskan penyelesaian masalah pada penelitian tugas akhir ini, maka batasan masalah yang diangkat adalah sebagai berikut:

- 1) Alat yang dirancang berfungsi untuk *monitoring level* pada tabung dan *flow* fluida pada *hydrostatic training system plant* skala laboratorium.
- 2) Data input sistem *monitoring* didapat dari 2 buah sensor ultrasonik HC-SR04 dan 2 buah *water flow sensor* yang telah diolah pada mikrokontroler AVR Atmega 1284P.
- 3) Digunakan modul ESP8266 untuk mentransmisikan data dari AVR Atmega 1284P ke ke *database MySQL web server* sehingga dapat ditampilkan pada aplikasi android yang dibuat menggunakan Android Studio.

1.4 Tujuan

Tujuan dibuatnya rancang bangun alat ini adalah untuk dapat merancang sistem *monitoring* jarak jauh pada *hydrostatic training system plant* skala laboratorium.

1.5 Sistematika Pelaporan

Adapun sistematika pelaporan pada tugas akhir ini adalah sebagai berikut.

Bab I berisi latar belakang, perumusan masalah, batasan masalah, tujuan, dan sistematika pelaporan.

Bab II berisi dasar teori.

Bab III berisi metodologi.

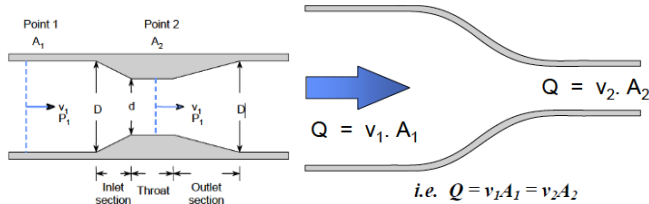
Bab IV berisi hasil dan pembahasan.

Bab V berisi kesimpulan dan saran.

Bab II Dasar Teori

2.1 Fluida

Fluida dibedakan menjadi dua, yaitu fluida cair dan gas. Partikel-partikel fluida yang bergerak disebut aliran fluida. Untuk mengetahui besar aliran fluida digunakan hukum kontinuitas Bernoulli ^[1] yang dapat dilihat pada gambar 2.1.



Gambar 2.1 Hukum Kontinuitas Bernoulli ^[1]

Berdasarkan gambar 2.1 di atas dapat diketahui bahwa tekanan aliran fluida dalam pipa (P) dipengaruhi oleh perbedaan kecepatan aliran fluida (v), debit aliran fluida (Q), dan perbedaan luas penampang pipa (A) pada *inlet* dan *outlet* pipa. Dengan luas penampang yang memiliki perbedaan diameter *inlet* (D) dan diameter *outlet* (d) pipa yang dapat dilihat pada persamaan ^[1] berikut.

$$\frac{v^2}{2} + \frac{P}{\rho} + gz = k \quad (2.1)$$

$$P_1 + \frac{1}{2} \cdot \rho \cdot v_1^2 = P_2 + \frac{1}{2} \cdot \rho \cdot v_2^2 \quad (2.2)$$

$$P_1 - P_2 = \frac{1}{2} \cdot \rho \cdot (v_2^2 - v_1^2) \quad (2.3)$$

$$\Delta P = \frac{1}{2} \cdot \rho \cdot (v_2^2 - v_1^2) \quad (2.4)$$

Dengan kecepatan aliran fluida $v_1 = \frac{Q}{A_1}$ dan $v_2 = \frac{Q}{A_2}$, sehingga didapat persamaan berikut:

$$\Delta P = \frac{1}{2} \cdot \rho \cdot \left\{ \left(\frac{Q}{A_1} \right)^2 - \left(\frac{Q}{A_2} \right)^2 \right\} \quad (2.5)$$

Sehingga debit aliran fluida (Q) menjadi:

$$Q = A_2 \sqrt{\frac{\frac{2\Delta P}{\rho}}{1 - \left(\frac{A_2}{A_1} \right)^2}} \quad (2.6)$$

Dengan luas penampang pipa $A_1 = \frac{\pi D^2}{4}$ dan $A_2 = \frac{\pi d^2}{4}$, sehingga didapat persamaan berikut:

$$Q = A_2 \sqrt{\frac{\frac{2\Delta P}{\rho}}{1 - \left(\frac{d}{D} \right)^2}} \quad (2.7)$$

Terdapat tiga jenis aliran fluida dalam pipa, yaitu aliran laminar, transisional dan turbulent yang dapat diketahui menggunakan bilangan Reynold (Re).

Berikut ini adalah jenis-jenis aliran fluida:

1) Aliran Laminar

Merupakan partikel-partikel fluida bergerak lancar dengan kecepatan rendah dan viskositas tinggi.

2) Aliran Transisional

Merupakan partikel-partikel fluida bergerak lancar kemudian berubah acak tidak stabil. Hal tersebut dikarenakan adanya perubahan kecepatan rendah ke tinggi dan perubahan viskositas tinggi ke rendah.

3) Aliran Turbulent

Merupakan partikel-partikel fluida bergerak acak tidak stabil dengan kecepatan tinggi dan viskositas rendah. ^[2]

Bilangan Reynold ^[1] didapat dari persamaan berikut:

$$Re = \frac{\rho v d}{\mu} \quad (2.8)$$

Selain bilangan Reynold (Re), adanya *head loss* (hf) dan faktor gesekan Darcy (f) dapat mempengaruhi jenis aliran fluida yang diketahui dengan persamaan Darcy ^[3] dan diagram Moody. ^[2]

Head loss (h_f) merupakan kerugian atau pengurangan aliran fluida disebabkan oleh faktor gesekan Darcy (f) yang besar.

Faktor gesekan Darcy (f) merupakan besar nilai gaya gesek pada permukaan dalam pipa yang berlawanan arah dengan arah aliran fluida. Besar nilai *head loss* (h_f) didapatkan dengan persamaan Darcy ^[3] berikut:

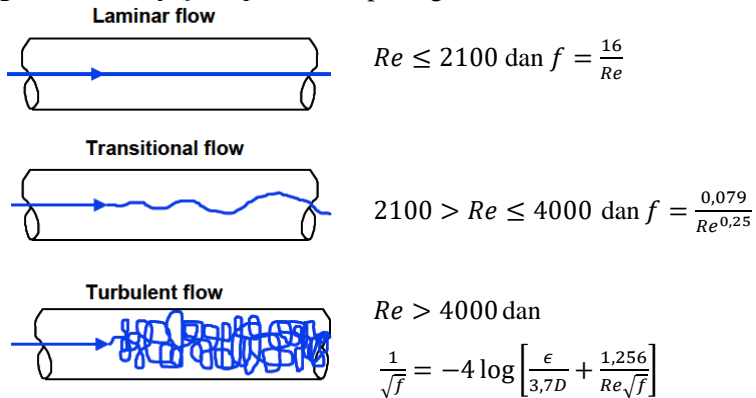
$$P_1 - P_2 = \rho g h_f \quad (2.9)$$

$$h_f = f \cdot \left(\frac{A_s}{A_c} \right) \cdot \frac{v^2}{2g} \quad (2.10)$$

Dengan $\left(\frac{A_s}{A_c} \right) = \frac{\pi D L}{\frac{\pi D^2}{4}} = \frac{4L}{D}$, sehingga didapatkan:

$$h_f = 4f \frac{L}{D} \cdot \frac{v^2}{2g} \quad (2.11)$$

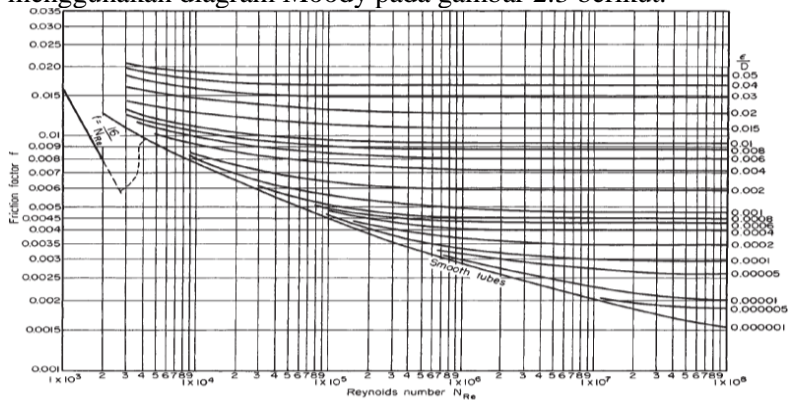
Perbedaan aliran fluida yang ditinjau dari besar nilai faktor gesekan Darcy (f) dapat dilihat pada gambar 2.2 berikut.



Gambar 2.2 Perbedaan Aliran Fluida ^{[1] [3]}

Berdasarkan gambar 2.2 tersebut dapat diketahui bahwa terdapat penentuan nilai faktor gesekan Darcy (f) sesuai dengan bilangan Reynold. Semakin besar nilai bilangan Reynold maka semakin kompleks persamaan Darcy nya.

Untuk mengetahui nilai faktor gesekan Darcy (f) dapat menggunakan diagram Moody pada gambar 2.3 berikut.



Gambar 2.3 Diagram Moody ^[2]

Berdasarkan gambar 2.3 di atas dapat diketahui bahwa nilai faktor gesekan Darcy ^[2] didapat dari:

- 1) Ditentukan nilai $\frac{\epsilon}{D}$ dan melihat bagian kanan diagram.
- 2) Ditarik garis ke kiri mengikuti garis melengkung.
- 3) Didapatkan nilai faktor gesekan Darcy (f) dilihat di bagian kiri diagram.
- 4) Didapatkan nilai bilangan Reynold (Re) dilihat di bagian bawah diagram.

Keterangan notasi dan satuan:

P = tekanan (Pa)

v = kecepatan aliran fluida (m/s)

ρ = densitas fluida (kg/m^3)

μ = viskositas fluida ($\text{Pa} \cdot \text{s}$)

d = diameter dalam pipa (m)

D = diameter luar pipa (m)

g = percepatan gravitasi (m/s^2)

z = level posisi

k = konstan

A = luas penampang (m^2)
 A_s = luas penampang dalam pipa (m^2)
 A_c = luas penampang luar pipa (m^2)
 h_f = *level* pada faktor Darcy
 f = faktor gesekan Darcy
 π = phi = 3,14
 L = panjang pipa (m)
 Re = bilangan Reynolds

2.2 Tekanan Hidrostatik

Tekanan Hidrostatik merupakan tekanan yang dipengaruhi oleh *level* fluida. Arah gaya tekan yang ke semua arah menghasilkan persamaan berikut.

$$P = \rho gh \quad (2.12)$$

Dimana perbedaan tekanan hidrostatik ΔP pada ketinggian berbeda ^[3] didapat persamaan:

$$\Delta P = \rho g \Delta h \quad (2.13)$$

Keterangan notasi dan satuan:

P = tekanan (Pa)

P_0 = tekanan mula – mula (Pa)

ρ = densitas fluida (kg/m^3)

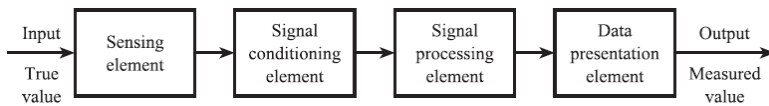
g = percepatan gravitasi (m/s^2)

h = level fluida (m)

Δh = perbedaan level fluida (m)

2.3 Sistem Monitoring

Sistem *monitoring* berfungsi memastikan proses sesuai harapan dan mengetahui perkembangan pencapaian target. Ada berbagai bentuk *monitoring*, yaitu dengan observasi, dokumentasi dan pembuatan aplikasi visual.^[4] Sistem *monitoring* pada *Hydrostatic Training System Plant* ini termasuk ke dalam bentuk visual secara *mobile* pada android yang diagram blok sistem nya dapat dilihat pada gambar 2.4 berikut.



Gambar 2.4 Diagram Blok Sistem *Monitoring* [5]

Berdasarkan gambar 2.4 di atas dapat diketahui bahwa:

- 1) *Input*
Merupakan nilai sebenarnya dari variabel fisis yang diukur.
Misalnya : variabel *level, flow, pressure, temperature* dsb.
- 2) *Sensing Element*
Merupakan perangkat pendeteksi variabel fisis, meneruskan data analog ke perangkat pengkondisi dan pemrosesan sinyal.
Misalnya: sensor ultrasonik, sensor mpx, thermocouple dsb.
- 3) *Signal Conditioning Element*
Merupakan perangkat pengkondisi sinyal *input* analog dari variabel fisis.
Misalnya: *oscillator* mengubah besaran impedansi ke frekuensi tegangan dsb.
- 4) *Signal Processing Element*
Merupakan perangkat pemrosesan sinyal *input* analog menjadi sinyal *output* yang diinginkan.
Misalnya: *Analogue to Digital Converter (ADC)* mengubah besaran tegangan analog menjadi digital input komputer.
- 5) *Data Presentation Element*
Merupakan perangkat untuk menampilkan data yang telah diukur dan diolah.
Misalnya : diagram data yang disimpan, *visual display unit* pada LCD, PC, android dsb.
- 6) *Output*
Merupakan nilai keluaran yang diinginkan.
Misalnya: misalnya besar nilai *level* yang diukur, posisi objek pada titik tertentu dan lain sebagainya.

2.4 Sensor

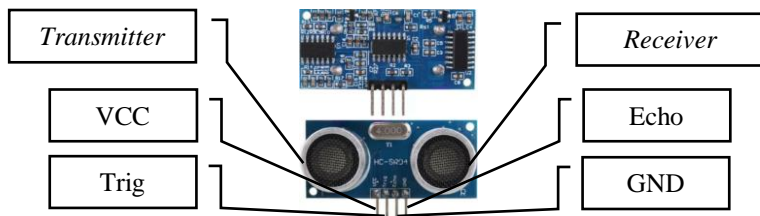
Sensor merupakan perangkat pendeteksi variabel fisis yang kemudian meneruskan data analog kepada perangkat pengkondisi dan pemrosesan sinyal. ^[5]

2.4.1 Sensor Ultrasonik

Sensor ultrasonik HC-SR04 adalah sensor pengukur jarak yang bekerja berdasarkan prinsip pantulan gelombang bunyi untuk mendeteksi keberadaan suatu objek ^[6] dengan spesifikasi dapat dilihat pada tabel 2.1 berikut.

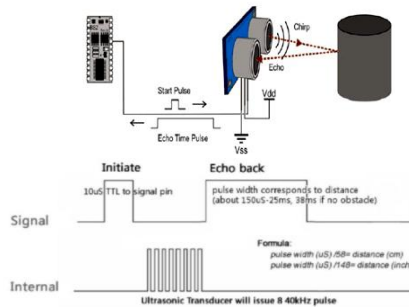
Tabel 2.1 Spesifikasi Sensor Ultrasonik

Keterangan	Nilai
Model	HC-SR04
Frekuensi	20 kHz-2 MHz
Power Supply	+5 VDC
Arus Kerja	15 mA
Range Pengukuran	2 cm – 400 cm
Resolusi	0.3 cm
Sudut Ukur	30°
Sinyal Input Trigger	10 μ S



Gambar 2.5 Konfigurasi Pin Sensor Ultrasonik HC-SR04 ^[7]

Berdasarkan gambar di atas dapat diketahui bahwa :
VCC sebagai *supply* sebesar +5 VDC
Trig (trigger) sebagai *input* sinyal ultrasonik ke benda
Echo sebagai *output* sinyal ultrasonik dari benda
GND sebagai *ground* ^[7]



Gambar 2.6 Cara Kerja Sensor Ultrasonik HC-SR04 [7]

Berdasarkan gambar cara kerja sensor Ultrasonik HC-SR04 di atas dapat diketahui bahwa pin *trig* (trigger) mendapatkan sinyal input (high) sebesar +5VDC dari pin *VCC* menghasilkan 40 kHz sinyal ultasonik. Bagian *transmitter* memancarkan sinyal ultrasonikk tersebut selama 10 μ s TTL menuju benda yang diukur jaraknya menghasilkan 8 cycle *stonic burst*. Benda tersebut memantulkan sinyal ultrasonik menuju ke bagian *receiver* dan diteruskan ke pin *echo* sehingga diterima sinyal output (high) +5VDC dan dikonversikan ke dalam satuan senti meter (*cm*).

Berikut adalah persamaan [6] untuk menentukan jarak:

- 1) Dalam satuan senti meter (*cm*)

$$S = \frac{t}{58} \quad (2.14)$$

- 2) Dalam satuan inchi (*in*)

$$S = \frac{t}{148} \quad (2.15)$$

- 3) Dalam satuan meter (*m*)

$$S = v \frac{t}{2} \quad (2.16)$$

Keterangan notasi dan satuan :

S = jarak objek

t = panjang pulsa *echo* (μ s)

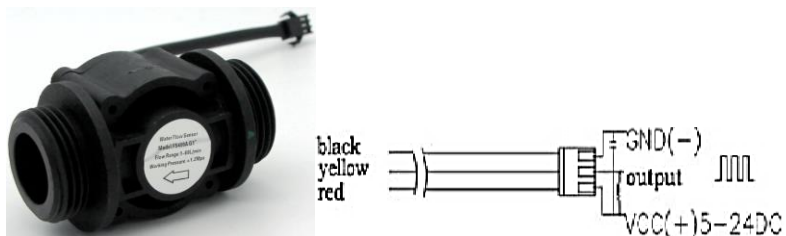
v = cepat rambat gelombang bunyi = 340 (m/s)

2.4.2 Water Flow Sensor

Water Flow Sensor adalah sensor untuk mengukur laju aliran fluida yang bekerja berdasarkan prinsip *hall effect*.^[8]

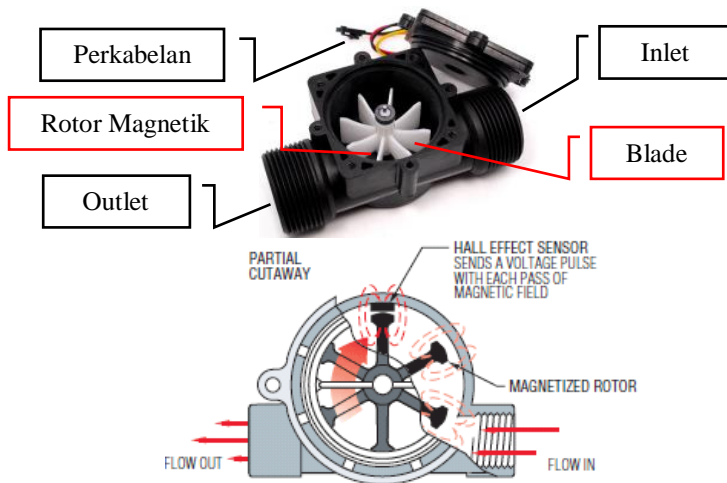
Tabel 2.2 Spesifikasi Water Flow Sensor

Keterangan	Nilai
Model	FS400A
Diameter	1 inch
Tegangan Kerja	5~24VDC
Tegangan Kerja Min	4.5 VDC
Tegangan Kerja Max	5 VDC
Arus Max	15mA
Range Laju Aliran	1~60L/min
Temperatur Cairan	≤120°C
Temperatur Operasi	≤80°C
Kelembaban Operasi	90%RH
Tekanan Air	≤1.75MPa
Tekanan Kerja	≤1.2MPa



Gambar 2.7 Konfigurasi Kabel Water Flow Sensor^[9]

Berdasarkan gambar 2.7 di atas dapat diketahui bahwa:
Kabel hitam *GND* sebagai *ground*
Kabel kuning *output* sebagai *output* sinyal
Kabel merah *VCC* sebagai *input* sinyal sebesar 5~24 VDC^[9]



Gambar 2.8 Cara Kerja Water Flow Sensor ^[8]

Berdasarkan gambar cara kerja *water flow sensor* di atas dapat diketahui bahwa ketika aliran fluida masuk ke dalam inlet sensor maka kabel merah *VCC* mendapatkan *input* sinyal sebesar 5 ~24 VDC. Aliran fluida berjalan mengenai rotor magnetik. Masing-masing blade rotor berinteraksi satu sama lain menghasilkan induksi magnetik yang dapat menggerakkan rotor sesuai arah jarum jam. Putaran rotor magnetik tersebut menghasilkan sinyal *output* yang sama dengan sinyal *input*. Pada kabel kuning sinyal *output* diteruskan ke mikrokontroler untuk dikonversi menjadi satuan liter per detik (L/s).^[8]

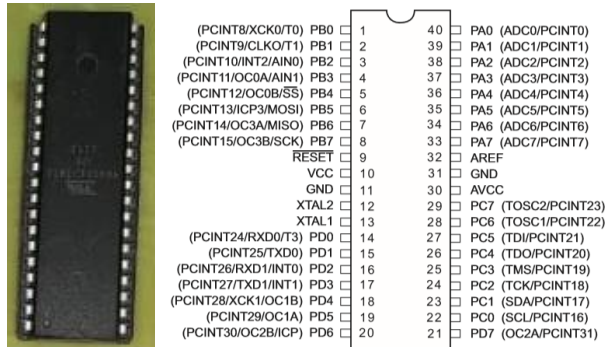
2.5 Mikrokontroler AVR Atmega 1284P

Mikrokontroler AVR Atmega 1284P adalah perangkat pengkondisian dan pemrosesan sinyal input sensor yang digunakan pada tugas akhir ini. Mikrokontroler AVR Atmega 1284P merupakan mikrokontroler 8 bit dengan 32 pin I/O, 4 KB EEPROM, dan 128 KB Flash Memory dapat diprogram ulang ^[10] yang spesifikasinya dapat dilihat pada tabel 2.3 berikut.

Tabel 2.3 Spesifikasi AVR Atmega 1284P

Keterangan	Nilai
Model	1284P
Bit	8
RISC Architecture	
Single Clock	131
Working Register	32 x 8 General Purpose
MIPS	20 at 20 mHz
Programable Memory	
Flash Memory	128 KB
EEPROM	4 KB
SRAM	16 KB
Peripheral Features	
PWM	6 Channel
ADC	8 Channel, 10 bit
Serial Interface	Master/Slave SPI
Serial USART	2 Programable
I/O	
Programable I/O Lines	32
PDIP	40 pin
Speed Grades	
0-4 MHz	1.5-5.5 V
0-10 MHz	2.7-5.5 V
0-20 MHz	4.5-5.5 V
Power Consumption at 1 MHz, 1.8 V, 25°C	
Active	0.4 mA

Gambar berikut ini adalah gambar konfigurasi pin AVR Atmega 1284P.



Gambar 2.9 Konfigurasi Pin AVR Atmega1284P ^[10]

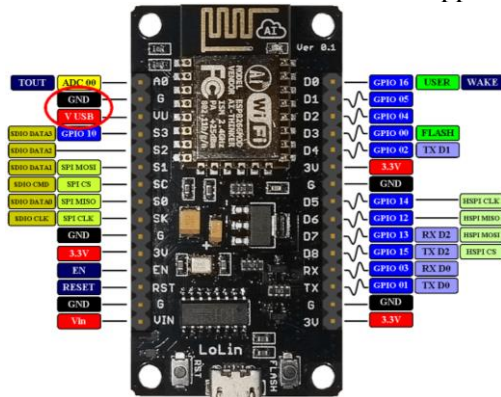
2.6 Modul Wifi ESP8266

Transmitter merupakan perangkat keras untuk menyalurkan data dari mikrokontroler pemroses dan pengkodisi sinyal agar dapat diolah pada sistem selanjutnya. ^[5] *Transmitter* yang digunakan adalah modul WiFi ESP8266 ^[11] untuk mengirimkan data dari Atmega1284P ke *database* MySQL yang spesifikasinya pada tabel 2.4 berikut.

Tabel 2.4 Spesifikasi Modul WiFi ESP8266

Keterangan	Nilai
Voltage	3,3 Volt
Wi-Fi Direct	(P2P) soft-AP
Current consumption	10 μ A~170 mA.
Flash memory	16 MB
Processor / speed	Tensilica L106 32-bit / 80~160 MHz
RAM	32K + 80K
GPIOs	17 (multiplexed with other functions)
ADC	1 input with 1024 step resolution
Output Power	+19.5dBm in 802.11b mode
802.11 support b/g/n	Integrated TCP/IP protocol stack
Max TCP connections	5

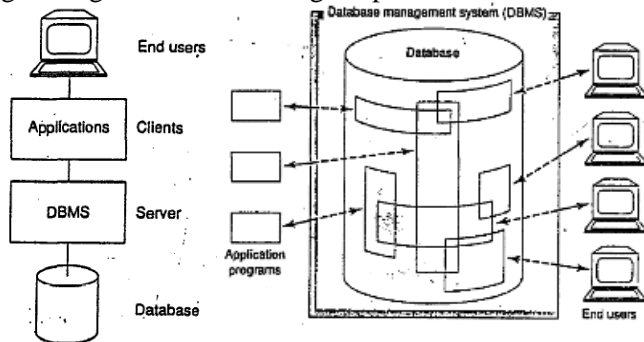
Secara umum konstruksi ESP8266 terdiri dari *hardware* dengan komunikasi dan menampilkan data sehingga dapat diakses melalui alamat IP dan *server* dan melakukan koneksi *AT Instruction Set*, *Cloud Server*, dan android/iOS App. ^[12]



Gambar 2.10 Konfigurasi Pin ESP8266 ^[11]

2.7 Database

Database adalah kumpulan data yang disimpan pada ruang penyimpanan elektronik. *Database* dapat diintegrasikan dan disebarkan ke seluruh *user*. Terdapat *Database Management System (DBMS)*, yaitu *software* untuk menjalankan seluruh akses dan menghubungkan *database* dengan aplikasi. ^[13]



Gambar 2.11 Arsitektur dan Komunikasi Client-Server ^[13]

Berdasarkan gambar tersebut dapat diketahui bahwa pada sistem arsitektur *client-server* semua *database* yang telah dibuat masuk ke DBMS *server*. DBMS *server* mengatur tata letak, urutan. Kemudian masuk pada *applications* yang dimiliki oleh *client* sehingga dapat digunakan atau diubah oleh *end users*. DBMS yang digunakan adalah MySQL.

MySQL adalah sebuah implementasi dari sistem manajemen basis data relasional (*RDBMS*) yang didistribusikan secara gratis di bawah lisensi GPL (*General Public License*). SQL (*Structure Query Language*) adalah sebuah konsep pengoperasian basis data, terutama untuk pemilihan atau seleksi dari pemasukan data yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

Berikut adalah kelebihan MySQL.

- 1) Cross Platform, dapat berjalan pada semua OS.
- 2) Menggunakan SQL (*Structured Query Language*) yang mudah dipahami.
- 3) Gratis oleh Oracle.
- 4) Dapat menampung data dalam jumlah besar tanpa mengurangi kecepatan aksesnya.
- 5) Sistem keamanan data cukup handal.
- 6) Dapat menampung semua tipe data.
- 7) Dapat diintegrasikan dengan berbagai bahasa pemrograman.
- 8) Mampu menangani banyak user tanpa kendala. ^{[14] [15]}

Database MySQL pada tugas akhir ini dibuat pada *web hosting* gratis <https://www.000webhost.com/> dengan nama *hosting* myhyts.000webhostapp.com.

2.8 *Android*

Data presentation element merupakan perangkat untuk menampilkan data yang telah diukur dan diolah. ^[5] Pada tugas akhir ini digunakan *display* aplikasi pada android.

Android adalah *mobile OS (Operating System)* modifikasi dari Linux. Android dibuat oleh *startup developer* dengan nama yang sama yaitu Android, Inc dan dikembangkan Google mulai tahun 2005. Terdapat banyak versi android mulai dari versi 1.1, 1.5 Cupcake, 1.6 Donut, 2.0-2.1 Éclair, 2.2 Froyo, 2.3 Gingerbeard, 4.0-4.3 Jelly Bean, 4.4 KitKat, 5.0-5.1 Lollipop, 6.0 Marshmallow, 7.0 Nougat, dan yang paling baru versi 8.0 Oreo.

Android memiliki fitur umum sebagai berikut:

- 1) Ruang Penyimpanan Data SQLite database.
- 2) Konektifitas GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, WiFi, LTE.
- 3) Perpesanan SMS dan MMS.
- 4) *Web browser open source* Chrome's V8 JavaScript engine.
- 5) Multimedia MP4, AMR, MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF, dan BMP.
- 6) *Hardware* didukung Sensor Accelerometer, Kamera, Kompas Digital, Sensor Proximity, dan GPS.
- 7) *Multi touch screens* dan *multi tasking*.
- 8) Android 2.3 supports Flash 10.1 dan tethering untuk berbagi *wired/wireless hotspot*.

2.9 Android Studio

Android Studio merupakan IDE developer aplikasi android yang menggunakan bahasa pemrograman Java.

Berikut ini adalah kelebihan dari Android Studio.

- 1) Menggunakan *Gradle-based build system* yang fleksibel.
- 2) Bisa mem-build multiple APK .
- 3) *Template* bisa untuk berbagai macam tipe perangkat.
- 4) *Layout editor* yang lebih bagus.
- 5) *Built-in* bisa untuk *Google Cloud Platform*, sehingga mudah untuk integrasi dengan *Google Cloud Messaging* dan *App Engine*.
- 6) *Import library* langsung dari *Maven repository*. ^{[16][17][18]}



Gambar 2.12 Android Studio

2.10 Karakteristik Statik

Karakteristik statik instrumen merupakan karakter yang menggambarkan parameter dari sebuah instrumen pada saat keadaan *steady*. Karakteristik ini tidak bergantung pada waktu.

Berikut ini adalah karakteristik statik:

1) *Range*

Range menyatakan jangkauan pengukuran sebuah alat ukur. *Range input* atau *output* sebuah elemen ditentukan dengan nilai minimum dan nilai maksimum dari *input* (*I*) atau *output* (*O*).

Dituliskan : I_{min} hingga I_{max} atau O_{min} hingga O_{max}

2) *Span*

Span merupakan selisih nilai maksimum dengan nilai minimum, menyatakan lebar daerah pengukuran.

Dituliskan : $I_{max} - I_{min}$ atau : $O_{max} - O_{min}$

3) *Linieritas*

Linearitas merupakan grafik yang menampilkan pengukuran nilai sebenarnya (*input*) menghasilkan *output* nilai yang ditunjukkan oleh instrumen (*output*).

Pengukuran dapat dikatakan ideal saat hubungan antara *input* pengukuran (nilai sesungguhnya) dengan *output* pengukuran (nilai yang ditunjukkan oleh alat) berbanding lurus.

Garis lurus ideal adalah garis menghubungkan titik minimum dengan titik maksimum *input/output*.

Berikut adalah persamaan linieritas.

$$O_{ideal} = KI + a \quad (2.17)$$

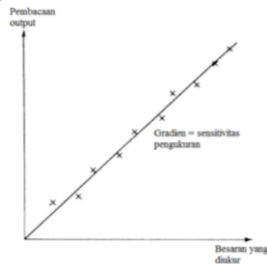
Dengan K adalah kemiringan garis, dapat diketahui melalui persamaan berikut.

$$K = \frac{O_{max} - O_{min}}{I_{max} - I_{min}} \quad (2.18)$$

Dan a adalah pembuat nol (*zero bias*), dapat dihitung dengan persamaan berikut.

$$a = O_{min} - KI_{min} \quad (2.19)$$

Berikut adalah grafik karakteristik statik linier instrumen:



Gambar 2.13 Karakteristik Linier Output Instrumen

4) Non – Linieritas

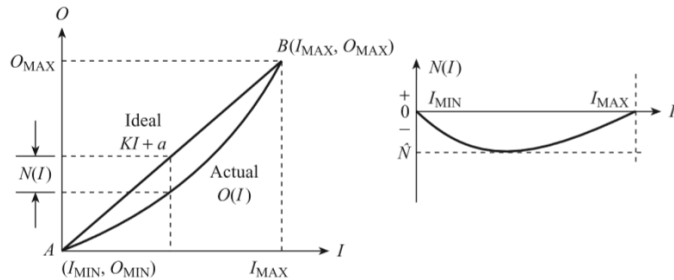
Dalam beberapa keadaan, dari persamaan linieritas muncul garis yang tidak lurus yang biasa disebut non-linier atau tidak linier. Didalam fungsi garis yang tidak linier ini menunjukkan perbedaan antara hasil pembacaan actual atau nyata dengan garis lurus idealnya.

Berikut adalah persamaan non linieritas.

$$N(I) = O(I) - (KI + a) \quad (2.20)$$

Sedangkan untuk persamaan dalam prosentase dari defleksi skala penuh, sebagai berikut.

$$N = \frac{\check{N}}{O_{max} - O_{min}} \times 100\% \quad (2.21)$$



Gambar 2.14 Grafik Non-Linieritas

5) Sensitivitas

Sensitivitas menunjukkan kepekaan sensor terhadap kuantitas yang diukur. Sensitivitas juga dapat diartikan sebagai perubahan *output* alat dibandingkan perubahan *input* dalam satu satuan.

Berikut adalah persamaan sensitivitas.

$$\text{Sensitivitas} = \frac{\Delta O}{\Delta I} \quad (2.22)$$

6) Resolusi

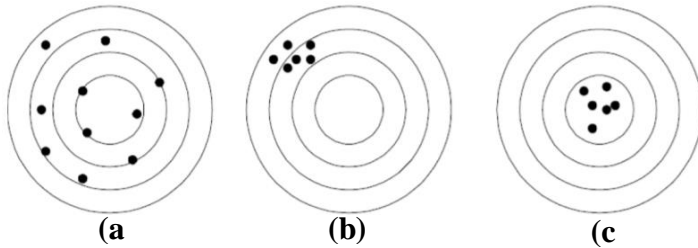
Resolusi merupakan perubahan terbesar *input* yang terjadi tanpa mempengaruhi perubahan *output*. Alat ukur dapat dikatakan mempunyai resolusi tinggi ketika mampu mengukur dengan ketelitian lebih kecil. Salah satu faktor utama yang mempengaruhi resolusi adalah seberapa baik skala *output* dibagi ke dalam skala lebih kecil.

7) Akurasi

Akurasi merupakan ketepatan alat ukur untuk memberikan nilai pengukuran mendekati nilai sebenarnya. Karena pada eksperimen nilai sebenarnya tidak pernah diketahui, oleh sebab itu diganti dengan suatu nilai standar yang diakui secara konvensional.

8) Presisi / Repeatability

Presisi adalah kemampuan instrument/element untuk menampilkan nilai yang sama pada pengukuran berulang singkat yang dapat dilihat pada gambar 2.15 berikut. ^[5]



Gambar 2.15 Hubungan Akurasi dan Presisi ^[5]

- (a) Akurasi rendah, presisi rendah
- (b) Akurasi rendah, presisi tinggi
- (c) Akurasi tinggi, presisi tinggi

2.11 Teori Ketidakpastian

Ketidakpastian pengukuran merupakan tingkat seberapa besar ketidakpastian yang dihasilkan oleh suatu alat ukur. Dalam menghitung ketidakpastian pengukuran ada beberapa langkah yang harus dihitung, antara lain:

1) Standard Deviasi

Berikut adalah persamaan standard deviasi.

$$\sigma = \frac{\sqrt{\sum (D_i - D')^2}}{n-1} \quad (2.23)$$

Keterangan:

D_i = koreksi alat ukur

D' = rata – rata koreksi

n = banyak range pengukuran

2) Analisa Tipe A (U_A)

Pada analisa tipe A ini hasilnya diperoleh dari data pengukuran.

Berikut adalah persamaan ketidakpastian analisa tipe A.

$$U_{A1} = \frac{\sigma_{max}}{\sqrt{n}} \quad (2.24)$$

Keterangan:

σ = standart deviasi

n = banyak range pengukuran

$$U_{A2} = \frac{SSR}{\sqrt{n-2}} \quad (2.25)$$

Dimana:

$$SSR = \sum SR \text{ (Square Residual)}$$

$$SR = R^2 \text{ (Residu)}$$

$$y_i = t_i - x_i \quad (2.26)$$

$$y_{reg} = a + (b \times t_i) \quad (2.27)$$

$$a = \bar{y}_i + (b \times \bar{t}_i) \quad (2.28)$$

$$b = \frac{n \cdot \sum t_i y_i - \sum y \cdot \sum t_i}{n \cdot \sum t_i^2 - (\sum t_i)^2} \quad (2.29)$$

Keterangan:

SSR = (Sum Square Residual)

t_i = Pembacaan standart

x_i = Pembacaan alat

y_i = Nilai koreksi

3) Analisa Type B (U_B)

Analisa tipe B ini diperoleh berdasarkan sertifikat kalibrasi atau spesifikasi dari alat ukur.

Berikut adalah persamaan ketidakpastian analisa tipe B.

$$U_{B1} = \frac{\text{Resolusi}/2}{\sqrt{3}} \quad (2.30)$$

$$U_{B2} = \frac{a}{k} \quad (2.31)$$

4) Ketidakpastian Kombinasi (U_C)

Merupakan ketidakpastian kombinasi dari ketidakpastian tipe A dan tipe B.

Berikut adalah persamaan ketidakpastian kombinasi.

$$U_C = \sqrt{U_{A1}^2 + U_{A2}^2 + U_{B1}^2 + U_{B2}^2} \quad (2.32)$$

Keterangan:

U_C = Nilai ketidakpastian kombinasi

U_{A1} = Nilai ketidakpastian hasil pengukuran

U_{A2} = Nilai ketidakpastian regresi

U_{B1} = Nilai ketidakpastian resolusi

U_{B2} = Nilai ketidakpastian kalibrator

5) V efektif

V efektif adalah derajat kebebasan efektif dari ketidakpastian kombinasi.

$$V_{eff} = \frac{(U_C)^4}{\sum (U_i)^4 / V_i} \quad (2.33)$$

Keterangan:

V_{eff} = Derajat kebebasan efektif

V_i = Derajat kebebasan ketidakpastian ke i

U_i = Hasil ketidakpastian analisa tipe A dan B

6) Faktor Cakupan (k)

Setelah ditentukan nilai derajat kebebasan efektif, maka dapat dilanjutkan dengan menghitung nilai faktor cakupan sesuai dengan tingkat kepercayaan yang diinginkan, dimana faktor cakupan (k) didapat dari tabel T-students.

7) Ketidakpastian Diperluas

Ketidakpastian diperluas merupakan akhir nilai ketidakpastian dengan tingkat kepercayaan. Tingkat kepercayaan tingkat keyakinan mengenai daerah nilai sebenarnya pada suatu pengukuran.

$$U_{exp} = k \cdot U_C \quad (2.34)$$

Untuk mencari nilai k , maka melihat table T student sesuai dengan confidence level 95%.

Halaman ini memang dikosongkan

Bab III

Metodologi

3.1 Alat dan Bahan

Berikut adalah alat dan bahan yang digunakan pada rancang bangun system *monitoring* jarak jauh *hydrostatic training system* skala laboratorium.

3.1.1 Sensor Ultrasonik HC-SR04

Sensor ultrasonik HC-SR04 adalah sensor pengukur jarak yang bekerja berdasarkan prinsip pantulan gelombang bunyi untuk mendeteksi keberadaan suatu objek. Pada tugas akhir ini digunakan untuk mengukur *level* fluida yang terletak pada tutup tabung A dan B yang dapat dilihat pada gambar 3.1 berikut.



Gambar 3.1 Sensor Ultrasonik HC-SR04

3.1.2 Water Flow Sensor

Water Flow Sensor adalah sensor untuk mengukur debit fluida yang bekerja berdasarkan prinsip *hall effect*. Pada tugas akhir ini digunakan untuk mengukur debit fluida yang terletak setelah *MOV* pada sistem A dan setelah pompa pada sistem B yang dapat dilihat pada gambar 3.2 berikut.



Gambar 3.2 Water Flow Sensor^[7]

3.1.3 AVR Atmega 1284P

Mikrokontroler AVR Atmega 1284P adalah perangkat pengkondisian dan pemrosesan sinyal *input* sensor yang digunakan pada tugas akhir ini dan dapat dilihat pada gambar 3.3 berikut.



Gambar 3.3 AVR Atmega1284P

3.1.4 Modul Wifi ESP8266

Transmitter yang digunakan adalah modul WiFi ESP8266 untuk mengirimkan data dari Atmega1284P ke *database* MySQL yang dapat dilihat pada gambar 3.4 berikut.



Gambar 3.4 ESP8266

3.1.5 Database MySQL

Database MySQL pada tugas akhir ini dibuat pada *web hosting* gratis <https://www.000webhost.com/> dengan nama *hosting* myhyts.000webhostapp.com.

3.1.6 Android Studio

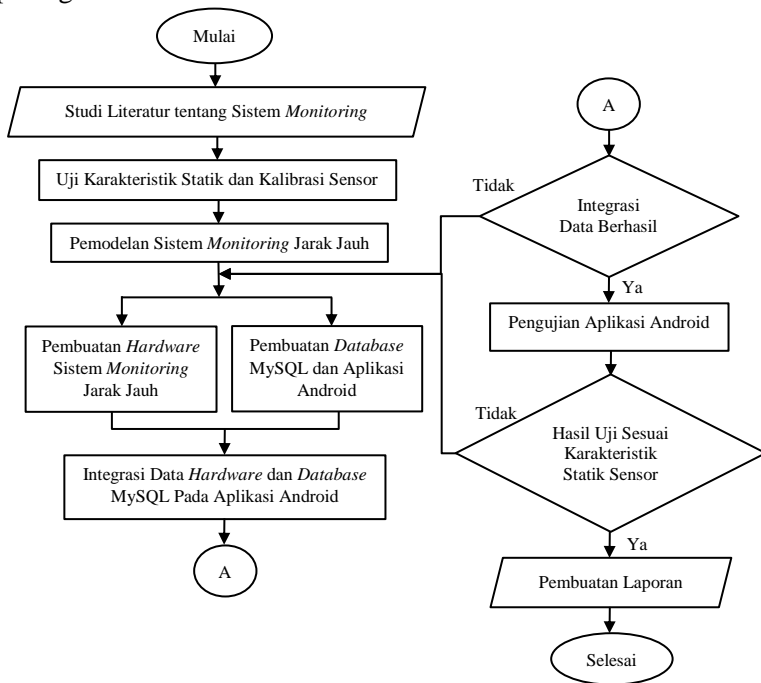
Android Studio merupakan IDE developer aplikasi android yang menggunakan bahasa pemrograman Java.



Gambar 3.5 Android Studio

3.2 Prosedur Percobaan

Pada proses pembuatan alat system *monitoring* jarak jauh pada *hydrostatic training system* skala laboratorium ini diperlukan adanya prosedur percobaan yang diimplementasikan dalam bentuk diagram alir yang berisi seluruh langkah pengerjaan dapat dilihat pada gambar 3.6 berikut ini.



Gambar 3.6 Diagram Alir Pembuatan Tugas Akhir

Berikut ini adalah penjelasan diagram alir pembuatan tugas akhir rancang bangun sistem *monitoring* jarak jauh pada *hydrostatic training system* skala laboratorium.

3.2.1 Studi Literatur

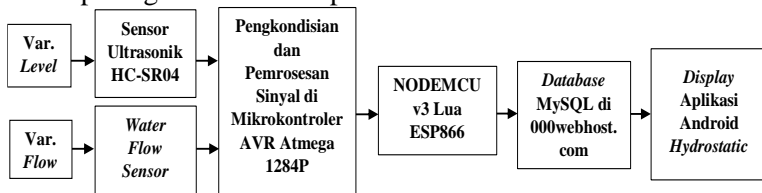
Studi literatur dilakukan untuk mendapatkan materi penunjang berasal dari buku dan jurnal media cetak maupun online.

3.2.2 Uji Karakteristik Statik dan Kalibrasi

Uji karakteristik statik dan kalibrasi dilakukan untuk mengetahui nilai karakteristik statik sensor yang dapat mempengaruhi tingkat akurasi alat dan hasil *monitoring*.

3.2.3 Identifikasi dan Pemodelan Sistem *Monitoring*

Proses identifikasi dan pemodelan sistem *monitoring* dapat dilihat pada gambar 3.7 sampai 3.10 berikut.



Gambar 3.7 Diagram Blok Sistem *Monitoring* Jarak Jauh [5]

Berdasarkan gambar 3.7 di atas dapat diketahui bahwa sistem *monitoring* jarak jauh ini memiliki:

1) *Input*

Berupa variabel *level* dan *flow*.

2) *Sensing Element*

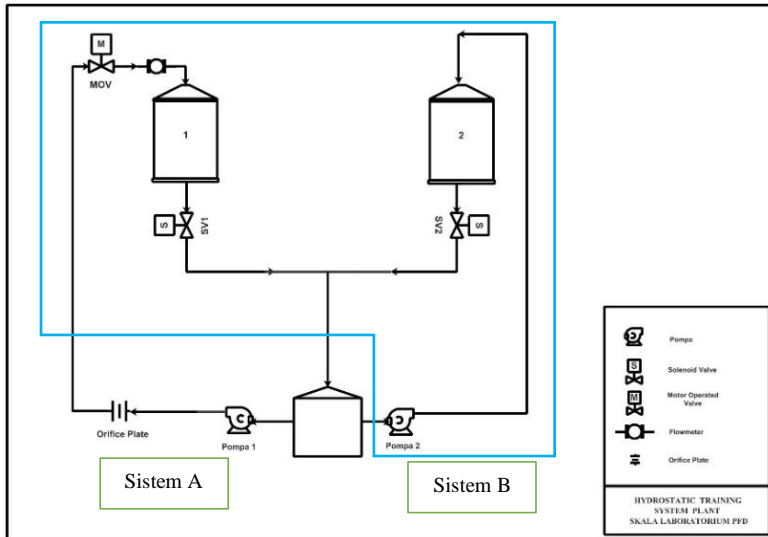
Berupa sensor ultrasonik HC-SR04 untuk mengukur *level* fluida dalam tabung A dan B dan *water flow sensor* untuk mengukur debit fluida pada sistem A dan B.

3) *Signal Conditioning & Processing Element*

Berupa Mikrokontroler AVR Atmega 1284P.

4) *Data Presentation Element*

Berupa aplikasi android untuk *display* jarak jauh.



**Gambar 3.8 PFD Sistem Monitoring Jarak Jauh Pada
*Hydrostatic Training System Skala Laboratorium***

Berdasarkan gambar 3.8 di atas dapat diketahui bahwa:

1) Sistem A

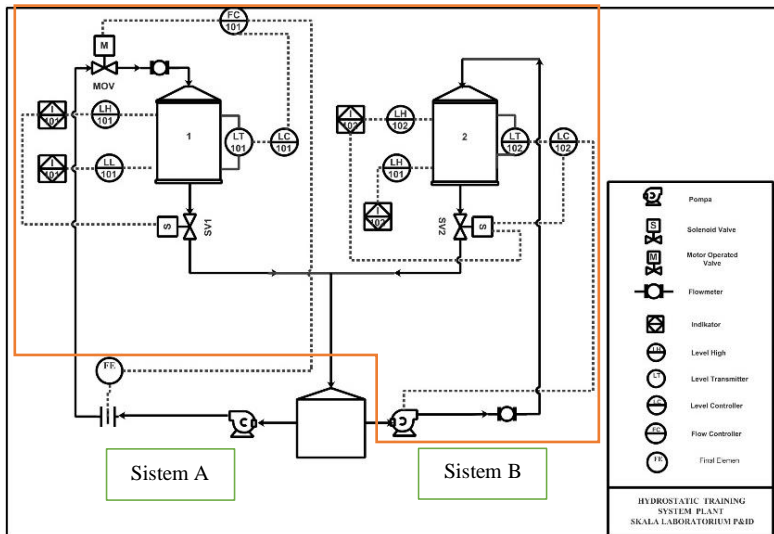
Sistem A merupakan sistem sebelah kiri, meliputi reservoir, pompa 1, *orifice plate*, *MOV*, *water flow sensor*, tabung 1, sensor ultrasonik HC-SR04 1, *solenoid valve* 1.

Data yang ditampilkan oleh sistem *monitoring* jarak jauh adalah hasil pembacaan *flow* dari *water flow sensor* yang terletak setelah *MOV* dan pembacaan *level* dari sensor ultrasonik HC-SR04 1 yang terletak pada tutup tabung 1.

2) Sistem B

Sistem B merupakan sistem sebelah kanan, meliputi reservoir, pompa 2, *water flow sensor*, tabung 2, sensor ultrasonik HC-SR04 2, *solenoid valve* 1.

Data yang ditampilkan oleh sistem *monitoring* jarak jauh adalah hasil pembacaan *flow* dari *water flow sensor* yang terletak setelah pompa 2 dan pembacaan *level* dari sensor ultrasonik HC-SR04 2 yang terletak pada tutup tabung 2.



Gambar 3.9 P&ID Sistem Monitoring Jarak Jauh Pada Hydrostatic Training System Skala Laboratorium

Berdasarkan gambar 3.9 di atas dapat diketahui bahwa:

- 1) Pompa 1 dan 2 dinyalakan dan *setting MOV, setpoint*, waktu simpan data di EEPROM mikrokontroler AVR Atmega 1284P ditentukan oleh sistem monitoring pada *plant*.
- 2) Pada Sistem A
Data hasil pembacaan *flow* dari *water flow sensor 1* sebagai *flow indicator* (FI 101) dalam satuan L/s, pembacaan *level* dari sensor ultrasonik HC-SR04 1 sebagai *level indicator* (LI 101) dalam satuan cm.
- 3) Pada Sistem B
Data hasil pembacaan *flow* dari *water flow sensor 2* sebagai *flow indicator* (FI 102) dalam satuan L/s, pembacaan *level* dari sensor ultrasonik HC-SR04 2 sebagai *level indicator* (LI 102) dalam satuan cm.

- 4) Pada sistem *monitoring* jarak jauh ini digunakan modul wifi ESP8266 sebagai *transmitter* data pembacaan *water flow sensor* dan sensor ultrasonik HC-SR04 di mikrokontroler AVR Atmega 1284P ke *database web server* sehingga dapat diakses pada aplikasi android.



Gambar 3.10 Alat *Hydrostatic Training System* Skala Laboratorium

3.2.4 Pembuatan *Hardware Sistem Monitoring Jarak Jauh*

Pembuatan *hardware* sistem *monitoring* dilakukan dengan meintegrasikan data dari sensor ultrasonik HC-SR04, *water flow sensor*, mikrokontroler AVR Atmega 1284P dan modul wifi ESP8266.

3.2.5 Pembuatan *Database MySQL dan Aplikasi Android*

Database MySQL pada tugas akhir ini dibuat pada *web hosting* gratis <https://www.000webhost.com/> dengan nama *hosting* myhyts.000webhostapp.com. Sedangkan *display* sistem *monitoring* jarak jauh ini berupa aplikasi android yang dibuat menggunakan Android Studio.

3.2.6 Integrasi Data *Hardware*, *Database* MySQL dan Aplikasi Android

Integrasi data *hardware*, *database* MySQL dan aplikasi android dilakukan agar sistem *monitoring* dapat berjalan secara otomatis dan lancar tanpa ada error. Jika integrasi data tidak berhasil maka dilakukan pembuatan sistem *monitoring* ulang.

3.2.7 Pembuatan Laporan

Laporan tugas akhir dibuat setelah memperoleh data karakteristik statik dan hasil *monitoring* yang tersimpan di *database* MySQL dalam aplikasi android. Data tersebut kemudian di analisa dan dimasukkan ke dalam laporan.

Bab IV

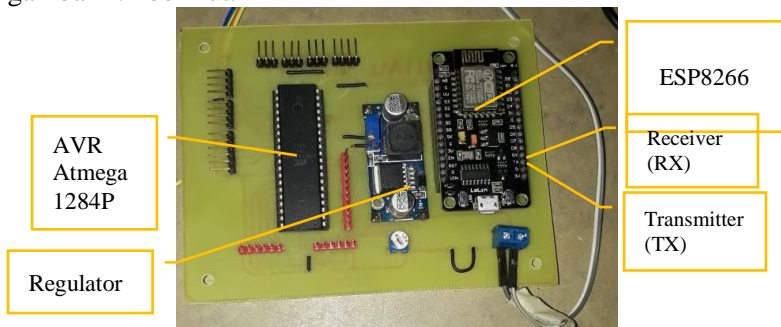
Hasil dan Pembahasan

4.1 Hasil Pembuatan Alat

Adapun data hasil pengujian dari tugas akhir rancang bangun sistem *monitoring* jarak jauh pada *hydrostatic training system* skala laboratorium ini adalah sebagai berikut.

4.1.1 Hardware Sistem Monitoring Jarak Jauh

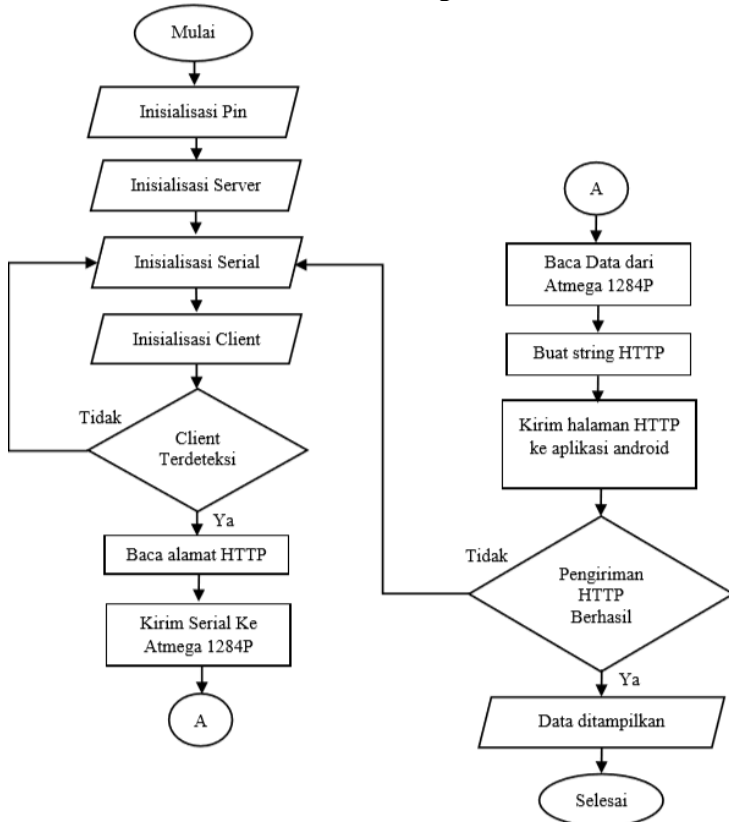
Hardware sistem *monitoring* jarak jauh berupa sensor ultrasonik HC-SR04, *water flow sensor*, mikrokontroler AVR Atmega 1284P dan modul wifi ESP8266 yang dapat dilihat pada gambar 4.1 berikut.



Gambar 4.1 ESP8266 Pada Board

Berdasarkan gambar 4.1 di atas dapat diketahui bahwa modul wifi ESP8266 mendapatkan input tegangan VCC dari PORTD5 mikrokontroler AVR Atmega 1284P. Karena nilai tegangan input VCC ESP8266 3.3 volt dan VCC Atmega 1284P 5 volt maka diperlukan adanya regulator untuk menurunkan tegangan Atmega 1284P agar data dapat ditransmisikan dengan baik tidak ada data *loses*. Komunikasi serial dari modul wifi ESP8266 ke Atmega 1284P menggunakan UART (TX) dan (RX) yaitu (TX) ESP8266 terhubung dengan (RX) Atmega 1284P.

Gambar 4.2 berikut adalah diagram alir serial ESP8266.



Gambar 4.2 Diagram Alir Serial ESP8266

Berikut ini adalah penjelasan diagram alir serial ESP8266.

1) Inisialisasi Pin

Dilakukan untuk mengetahui input data ESP8266 yaitu pada (TX) 12 byte dan (RX) 14 byte.

2) Inisialisasi *Server*

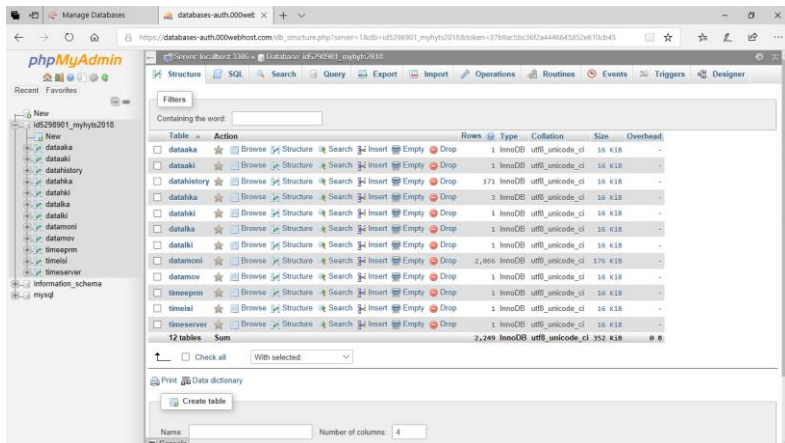
Dilakukan untuk mengetahui alamat *server HTTP* yang digunakan. Pada tugas akhir ini menggunakan *server* <https://www.myhyts.000webhost.com/>.

- 3) *Inisialisasi Serial*
Dilakukan untuk mentransmisikan data. ESP8266 ini menggunakan *baud rate* 9600 Hz.
- 4) *Inisialisasi Client*
Dilakukan agar client dapat terhubung dengan ESP8266. Client mendeteksi ESP8266 dengan nama perangkat “ESP_FD8818”, kemudian dihubungkan ke *hotspot portable HP* dengan *SSID* “Prime7” dan *password* “riha0895”.
- 5) *Baca Alamat HTTP*
Dilakukan agar alamat *web server* yang telah dibuat yaitu <https://www.myhyts.000webhost.com/> dapat dibaca oleh ESP8266 kemudian dikirim ke Atmega 1284P.
- 6) *Kirim Serial ke Atmega 1284P*
Dilakukan agar dapat mengakses data digital dari sensor dan aktuator yang tersimpan di Atmega 1284P dan terdisplay di LCD.
- 7) *Baca Data dari Atmega 1284P*
Data yang berbentuk digital yang sudah bisa diakses tersebut diubah dalam bentuk string agar dapat dikirimkan ke *database web server*.
- 8) *Buat String HTTP*
Pembuatan string untuk mempermudah pengiriman data dalam bentuk *HTTP* ke Atmega 1284P.

Sedangkan *listing* program mentransmisikan data dari Atmega 1284P ke *database web server* dan *setting* ESP8266 dapat dilihat pada lampiran A dan B.

4.1.2 Database MySQL

Database MySQL pada tugas akhir ini dibuat pada *web hosting* gratis <https://www.000webhost.com/> dengan nama *server* <https://www.myhyts.000webhost.com/>. Terdapat 12 tabel *database* yang telah dibuat dapat dilihat pada gambar 4.2 berikut.



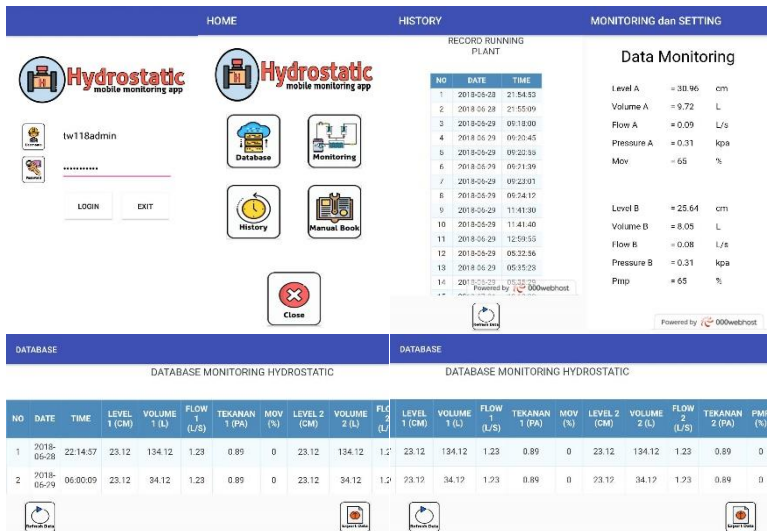
Gambar 4.3 Pembuatan Database MySQL

Berdasarkan gambar 4.2 di atas dapat diketahui bahwa:

- 1) *Database* yang digunakan adalah tipe InnoDB dan nama pengguna “id2598901_myhyts2018”.
- 2) Terdapat 12 tabel *database*, meliputi:
 - 7 sebagai tabel input per variable yang diukur (*dataaaka*, *dataaiki*, *datahka*, *datahki*, *dataalka*, *datalki*, *datamov*) yang ditampilkan pada tabel *monitoring*.
 - 1 tabel bernama *datamoni* berisi tabel *level*, volume, *flow*, bukaan MOV, tegangan pompa pada menu *database*.
 - 1 tabel bernama *datahistory* untuk menu *history* berisi tanggal dan waktu.
 - 3 Tabel *setting* waktu yaitu *timeeprm* untuk mengakses waktu penyimpanan data pada EEPROM, *timeisi* untuk mengetahui waktu kirim data, *timeserver* untuk mengirimkan data ke *database web server* setiap 5 detik.

4.1.3 Aplikasi Android

Aplikasi android yang dibuat menggunakan Android Studio dengan spesifikasi minimum OS android 6.0 Marshmallow memiliki tampilan sebagai berikut.



Gambar 4.4 Tampilan Aplikasi Android

Berdasarkan gambar 4.3 di atas aplikasi android memiliki 6 menu yaitu sebagai berikut.

- 1) *Login*, berisi *username* dan *password* untuk masuk ke dalam aplikasi.
Username “tw118admin” dan *password* “TAlancar123”.
- 2) *Home*, berisi 4 menu utama.
Yaitu *database*, *monitoring*, *history*, dan *manual book*.
- 3) *Database*, berisi *database monitoring*.
Pada menu ini *database MySQL* ditampilkan dalam bentuk *web view* dan terdapat fitur *ekspor database* ke dalam bentuk *tabel excel (.xls)*.
- 4) *Monitoring*, berisi *status plant* saat dijalankan.
Pada menu ini nilai akan berubah-ubah sesuai dengan nilai data yang masuk ke *database*.
- 5) *History*, berisi *riwayat plant* dijalankan.
Pada menu ini data yang terekam adalah data tanggal dan waktu *plant* dinyalakan.
- 6) *Manual Book*, berisi buku petunjuk penggunaan *plant*.

Sedangkan *listing* program aplikasi android pada Android Studio dapat dilihat pada lampiran C.

4.2 Data Hasil Pengujian Alat

Data hasil pengujian alat pada tugas akhir ini meliputi pengujian, karakteristik statik dan perbandingan pembacaan sensor.

4.2.1 Pengujian Sensor Ultrasonik HC-SR04

Pada penelitian tugas akhir ini menggunakan sensor ultrasonik HC-SR04 yang memiliki *range* 2 cm – 4 m pada *datasheet*. Berikut merupakan data pengukuran dari pengujian sensor ultrasonik.

Tabel 4.1 Hasil Pengujian Naik Sensor Ultrasonik HC-SR04

Jarak 5 cm				
Data Ke	Uji (cm)	Standar (cm)	Koreksi	Error (%)
1	5.51	5	-0.51	-10.2
2	5.38	5	-0.38	-7.6
3	6.12	5	-1.12	-22.4
4	5.64	5	-0.64	-12.8
5	5.83	5	-0.83	-16.6
Jarak 10 cm				
Data Ke	Uji (cm)	Standar (cm)	Koreksi	Error (%)
1	10.51	10	-0.51	-5.1
2	10.32	10	-0.32	-3.2
3	10.54	10	-0.54	-5.4
4	11.05	10	-1.05	-10.5
5	10.89	10	-0.89	-8.9
Jarak 15 cm				
Data Ke	Uji (cm)	Standar (cm)	Koreksi	Error (%)
1	15.43	15	-0.43	-2.866666667
2	15.52	15	-0.52	-3.466666667
3	15.62	15	-0.62	-4.133333333
4	15.64	15	-0.64	-4.266666667
5	15.45	15	-0.45	-3

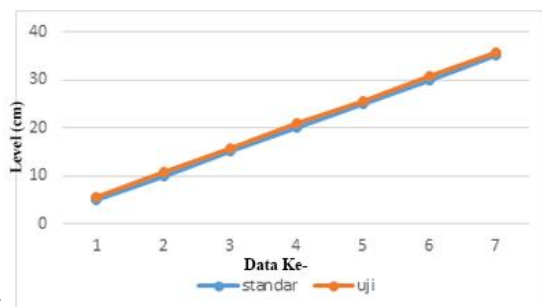
Jarak 20 cm				
Data Ke	Uji (cm)	Standar (cm)	Koreksi	Error (%)
1	20.53	20	-0.53	-2.65
2	20.56	20	-0.56	-2.8
3	21.1	20	-1.1	-5.5
4	21.05	20	-1.05	-5.25
5	20.65	20	-0.65	-3.25

Tabel 4.2 Hasil Pengujian Turun Sensor Ultrasonik HC-SR04

Jarak 35 cm				
Data Ke	Uji (cm)	Standar (cm)	Koreksi	Error (%)
1	35.21	35	-0.21	-0.6
2	35.55	35	-0.55	-1.571428571
3	36.05	35	-1.05	-3
4	36.02	35	-1.02	-2.914285714
5	35.88	35	-0.88	-2.514285714
Jarak 30 cm				
Data Ke	Uji (cm)	Standar (cm)	Koreksi	Error (%)
1	30.51	30	-0.51	-1.7
2	30.59	30	-0.59	-1.966666667
3	30.5	30	-0.5	-1.666666667
4	31.02	30	-1.02	-3.4
5	30.92	30	-0.92	-3.066666667
Jarak 25 cm				
Data Ke	Uji (cm)	Standar (cm)	Koreksi	Error (%)
1	25.56	25	-0.56	-2.24
2	25.7	25	-0.7	-2.8
3	25.05	25	-0.05	-0.2
4	25.91	25	-0.91	-3.64
5	25.89	25	-0.89	-3.56
Jarak 20 cm				
Data Ke	Uji (cm)	Standar (cm)	Koreksi	Error (%)
1	20.54	20	-0.54	-2.7
2	21.29	20	-1.29	-6.45
3	21.12	20	-1.12	-5.6
4	20.94	20	-0.94	-4.7
5	20.73	20	-0.73	-3.65

Jarak 15 cm				
Data Ke	Uji (cm)	Standar (cm)	Koreksi	Error (%)
1	15.54	15	-0.54	-3.6
2	16.03	15	-1.03	-6.866666667
3	15.78	15	-0.78	-5.2
4	15.88	15	-0.88	-5.866666667
5	15.73	15	-0.73	-4.866666667
Jarak 10 cm				
Data Ke	Uji (cm)	Standar (cm)	Koreksi	Error (%)
1	10.5	10	-0.5	-5
2	11.02	10	-1.02	-10.2
3	10.56	10	-0.56	-5.6
4	10.31	10	-0.31	-3.1
5	11.09	10	-1.09	-10.9
Jarak 5 cm				
Data Ke	Uji (cm)	Standar (cm)	Koreksi	Error (%)
1	5.21	5	-0.21	-4.2
2	5.49	5	-0.49	-9.8
3	5.56	5	-0.56	-11.2
4	5.64	5	-0.64	-12.8
5	5.38	5	-0.38	-7.6

Berdasarkan hasil pengujian sensor didapatkan grafik sebagai berikut.



Gambar 4.5 Grafik Perbandingan Alat Ukur Standar dan Uji sensor Ultrasonik HC-SR04

4.2.2 Karakteristik Sensor Ultrasonik HC-SR04

Karakteristik statik adalah karakteristik yang harus diperhatikan apabila alat tersebut digunakan untuk mengukur suatu kondisi yang tidak berubah karena waktu atau hanya berubah secara lambat laun.

Berikut adalah nilai karakteristik statik sensor Ultrasonik HC-SR04.

- a. Range : 2 cm – 400 cm
- b. Span : 398 cm
- c. Resolusi : 0,01 cm
- d. Sensitivitas : 1.00207 (dapat dilihat pada persamaan 4.1)

$$K(\text{Sensitivitas}) = \frac{\Delta O}{\Delta I} = \frac{35,638 - 5.576}{35 - 5} = 1.00207 \quad (4.1)$$

- e. *Non – Linieritas*

$$\hat{N} = \frac{[O - KI + a]_{\max}}{O_{\max} - O_{\min}} 100\% \quad (4.2)$$

Dimana :

$$a (\text{zero bias}) = O_{\min} - KI_{\min} \quad (4.3)$$

$$a = 5.576 - (1.00207)(5)$$

$$a = 0.56565$$

- f. Akurasi :

$$A = 1 - \left| \frac{Y_n - X_n}{Y_n} \right| \times 100\%, \quad (4.4)$$

$$A = 1 - \left| \frac{35 - 35.638}{35} \right| \times 100\%$$

$$A = 98,178\%$$

dengan Y_n = Pembacaan standar dan X_n = Pembacaan alat

- g. Error :

$$e = 1 - A \quad (4.5)$$

$$e = 1 - 0.98178$$

$$e = 0.01822 = 1,822\%$$

4.2.3 Perbandingan Pembacaan Sensor Pada Jarak Pancar ESP8266 4m dari *Plant*

Tabel 4.3 Data Pembacaan Sistem A Setpoint 25cm Di Android Pada Jarak Pancar ESP8266 4m dari *Plant*

Date	Time	LEVEL 1 (cm)	VOLUME 1 (L)	FLOW 1 (L/s)	TEKANAN 1 (p)	MOV (%)
7/17/2018	21:43:55	2.26	0.71	0.32	0.02	90
7/17/2018	21:43:58	2.28	0.72	0.47	0.02	90
7/17/2018	21:44:01	2.97	0.93	0.8	0.03	90
7/17/2018	21:44:04	4.13	1.3	0.8	0.04	90
7/17/2018	21:44:07	6.67	2.1	0.58	0.07	90
7/17/2018	21:44:09	10.21	3.21	0.58	0.1	90
7/17/2018	21:44:12	13.04	4.1	0.58	0.13	90
7/17/2018	21:44:15	15.57	4.89	0.58	0.16	90
7/17/2018	21:44:18	15.53	4.88	0.51	0.16	90
7/17/2018	21:44:20	18.81	5.91	0.51	0.19	90
7/17/2018	21:44:23	24	7.54	0.51	0.24	89
7/17/2018	21:44:26	25.18	7.91	0.27	0.25	89
7/17/2018	21:44:28	24.96	7.84	0.2	0.25	89
7/17/2018	21:44:31	24.29	7.63	0.2	0.24	89
7/17/2018	21:44:34	21.58	6.78	0.31	0.22	70
7/17/2018	21:44:37	25.66	8.06	0.31	0.26	89
7/17/2018	21:44:39	25.39	7.97	0.51	0.25	89
7/17/2018	21:44:42	25.93	8.14	0.18	0.26	89
7/17/2018	21:44:45	25.44	7.99	0.18	0.25	89
7/17/2018	21:44:48	25.44	7.99	0.18	0.25	89
7/17/2018	21:44:50	25.71	8.07	0.18	0.26	89
7/17/2018	21:44:53	25.09	7.88	0.18	0.25	89

Berdasarkan tabel 4.3 dapat diketahui bahwa data pembacaan sistem A dalam mencapai setpoint 25 cm di android pada jarak 4 m dari *plant* terjadi *delay* rata-rata pengiriman data selama 3 detik. Bukan MOV 89% saat *level* mencapai 25.18 cm dan *flow* 0.12 l/s.

**Tabel 4.4 Data Pembacaan Sistem B Setpoint 25cm Di Android
Pada Jarak Pancar ESP8266 4m dari *Plant***

Date	Time	LEVEL 2 (cm)	VOLUME 2 (L)	FLOW 2 (L/s)	TEKANAN 2 (p)	PMP (%)
7/17/2018	21:43:55	3.15	0.99	0.27	0.03	90
7/17/2018	21:43:58	2.72	0.85	0.37	0.03	90
7/17/2018	21:44:01	3.56	1.12	0.64	0.04	90
7/17/2018	21:44:04	5.8	1.82	0.64	0.06	90
7/17/2018	21:44:07	9.11	2.86	0.64	0.09	90
7/17/2018	21:44:09	12.51	3.93	0.55	0.13	90
7/17/2018	21:44:12	14.18	4.45	0.55	0.14	90
7/17/2018	21:44:15	15.48	4.86	0.55	0.15	90
7/17/2018	21:44:18	19.36	6.08	0.55	0.19	90
7/17/2018	21:44:20	21.44	6.73	0.45	0.21	70
7/17/2018	21:44:23	20.85	6.55	0.45	0.21	70
7/17/2018	21:44:26	24.63	7.73	0.3	0.25	50
7/17/2018	21:44:28	25.41	7.98	0.2	0.25	50
7/17/2018	21:44:31	24.16	7.59	0.2	0.24	50
7/17/2018	21:44:34	25.05	7.87	0.22	0.25	50
7/17/2018	21:44:37	22.65	7.11	0.29	0.23	70
7/17/2018	21:44:39	23.47	7.37	0.29	0.23	50
7/17/2018	21:44:42	24.68	7.75	0.13	0.25	50
7/17/2018	21:44:45	23.33	7.32	0.13	0.23	50
7/17/2018	21:44:48	25.41	7.98	0.11	0.25	50
7/17/2018	21:44:50	24.04	7.55	0.2	0.24	50
7/17/2018	21:44:53	24.98	7.84	0.11	0.25	50

Berdasarkan tabel 4.4 dapat diketahui bahwa data pembacaan sistem B dalam mencapai setpoint 25 cm di android pada jarak 4 m dari *plant* terjadi *delay* rata-rata pengiriman data selama 3 detik. Tegangan pompa 50% saat *level* mencapai 25.41 cm dan *flow* 0.12 l/s.

**Tabel 4.5 Data Pembacaan Sistem A Setpoint 25cm Di LCD
Pada Jarak Pancar ESP8266 4m dari Plant**

LEVEL 1 (cm)	VOLUME 1 (L)	FLOW 1 (L/s)	TEKANAN 1 (p)	MOV (%)
2.1	0.66	0.38	0.02	90
2.2	0.69	0.5	0.02	90
2.77	0.87	0.85	0.03	90
4	1.26	0.85	0.04	90
6.5	2.04	0.6	0.07	90
10.11	3.17	0.6	0.1	90
13	4.08	0.6	0.13	90
15.37	4.83	0.6	0.16	90
15.43	4.85	0.53	0.16	90
18.71	5.87	0.5	0.19	90
23.89	7.50	0.5	0.24	89
25.08	7.88	0.32	0.25	89
24.88	7.81	0.22	0.25	89
24.19	7.60	0.22	0.24	89
21.48	6.74	0.35	0.22	70
25.55	8.02	0.35	0.26	89
25.33	7.95	0.54	0.25	89
25.7	8.07	0.18	0.26	89
25.4	7.98	0.18	0.25	89
25.4	7.98	0.18	0.25	89
25.3	7.94	0.18	0.26	89
24.9	7.82	0.18	0.25	89

Berdasarkan tabel 4.5 dapat diketahui bahwa data pembacaan sistem A dalam mencapai setpoint 25 cm di LCD memiliki nilai tegangan pompa 50% saat *level* mencapai 25.41 cm dan *flow* 0.12 l/s.

**Tabel 4.6 Data Pembacaan Sistem B Setpoint 25cm Di LCD
Pada Jarak Pancar ESP8266 4m dari Plant**

LEVEL 2 (cm)	VOLUME 2 (L)	FLOW 2 (L/s)	TEKANA N 2 (p)	PMP (%)
3.12	0.98	0.29	0.03	90
2.62	0.82	0.42	0.03	90
3.36	1.06	0.72	0.04	90
5.6	1.76	0.72	0.06	90
9.08	2.85	0.7	0.09	90
12.41	3.90	0.6	0.13	90
14.08	4.42	0.6	0.14	90
15.38	4.83	0.6	0.15	90
19.32	6.07	0.6	0.19	90
21.34	6.70	0.49	0.21	70
20.7	6.50	0.49	0.21	70
24.43	7.67	0.32	0.25	50
25.32	7.95	0.22	0.25	50
24.06	7.55	0.22	0.24	50
23.9	7.50	0.31	0.25	50
22.45	7.05	0.35	0.23	70
23.37	7.34	0.35	0.23	50
24.48	7.69	0.13	0.25	50
23.13	7.26	0.13	0.23	50
25.21	7.92	0.11	0.25	50
23.8	7.47	0.2	0.24	50
24.58	7.72	0.11	0.25	50

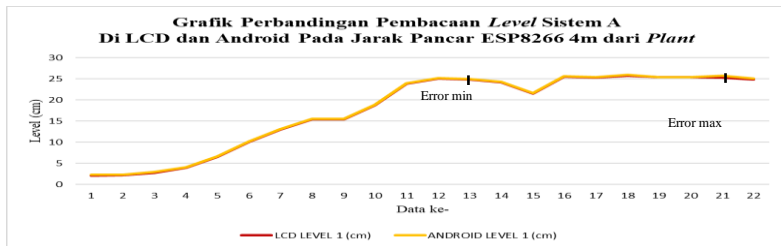
Berdasarkan tabel 4.6 dapat diketahui bahwa data pembacaan sistem B dalam mencapai setpoint 25 cm di LCD memiliki nilai tegangan pompa 50% saat *level* mencapai 25.32 cm dan *flow* 0.22 l/s.

Tabel 4.7 Perbandingan Data Pembacaan *Level* Setpoint 25 cm LCD dan Android Pada Jarak Pancar ESP8266 4 m dari *Plant*

LCD	ANDROID		LCD	ANDROID	
LEVEL 1 (cm)	LEVEL 1 (cm)	Error	LEVEL 2 (cm)	LEVEL 2 (cm)	Error
2.1	2.26	0.16	3.12	3.15	0.03
2.2	2.28	0.08	2.62	2.72	0.1
2.77	2.97	0.2	3.36	3.56	0.2
4	4.13	0.13	5.6	5.8	0.2
6.5	6.67	0.17	9.08	9.11	0.03
10.11	10.21	0.1	12.41	12.51	0.1
13	13.04	0.04	14.08	14.18	0.1
15.37	15.57	0.2	15.38	15.48	0.1
15.43	15.53	0.1	19.32	19.36	0.04
18.71	18.81	0.1	21.34	21.44	0.1
23.89	24	0.11	20.7	20.85	0.15
25.08	25.18	0.1	24.43	24.63	0.2
24.88	24.96	0.08	25.32	25.41	0.09
24.19	24.29	0.1	24.06	24.16	0.1
21.48	21.58	0.1	23.9	25.05	1.15
25.55	25.66	0.11	22.45	22.65	0.2
25.33	25.39	0.06	23.37	23.47	0.1
25.7	25.93	0.23	24.48	24.68	0.2
25.4	25.44	0.04	23.13	23.33	0.2
25.4	25.44	0.04	25.21	25.41	0.2
25.3	25.71	0.41	23.8	24.04	0.24
24.9	25.09	0.19	24.58	24.98	0.4

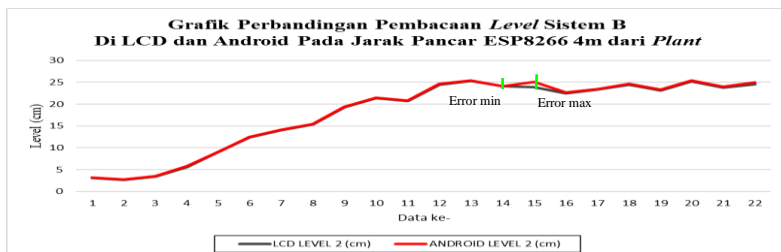
Berdasarkan tabel 4.7 dapat diketahui bahwa data pembacaan *level* dengan setpoint 25 cm di LCD dan android didapatkan nilai selisih pembacaan terbesar 0.41 cm dan terkecil 0.08 cm untuk *Level* 1 dan nilai selisih pembacaan terbesar 1.15 cm dan terkecil 0.03 cm untuk *Level* 2.

Sehingga didapatkan grafik sebagai berikut.



Gambar 4.6 Grafik Perbandingan Pembacaan *Level* Sistem A Di LCD dan Android Pada Jarak Pancar ESP8266 4m dari *Plant*

Berdasarkan gambar 4.6 dapat diketahui bahwa pembacaan data *level* 1 sistem A terdapat error maksimum pada data ke 21 memiliki selisih 0.41 cm yaitu di LCD terbaca 25.3 cm sedangkan di android 25.71 cm. Error minimum data ke 13 memiliki selisih 0.09 cm yaitu di LCD terbaca 24.88 cm sedangkan di android 24.96 cm.



Gambar 4.7 Grafik Perbandingan Pembacaan *Level* Sistem B Di LCD dan Android Pada Jarak Pancar ESP8266 4m dari *Plant*

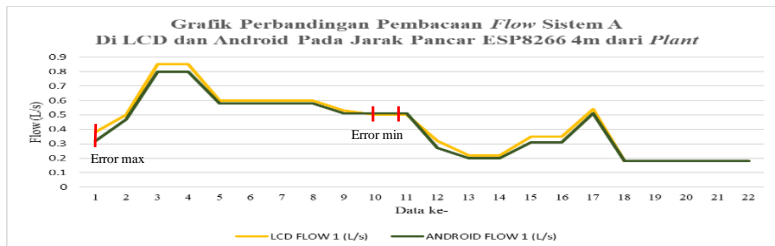
Berdasarkan gambar 4.7 dapat diketahui bahwa pembacaan data *level* 2 sistem B terdapat error maksimum pada data ke 15 memiliki selisih 1.15 cm yaitu di LCD terbaca 23.9 cm sedangkan di android 25.05 cm. Error minimum pada data ke 13 memiliki selisih 0.03 cm yaitu di LCD terbaca 25.32 cm sedangkan di android 25.41 cm.

**Tabel 4.8 Perbandingan Data Pembacaan *Flow* Setpoint 25 cm
LCD dan Android Pada Jarak Pancar ESP8266 4m
dari *Plant***

LCD	ANDROID		LCD	ANDROID	
FLOW 1 (L/s)	FLOW 1 (L/s)	Selisih	FLOW 2 (L/s)	FLOW 2 (L/s)	Selisih
0.38	0.32	-0.06	0.29	0.27	-0.02
0.5	0.47	-0.03	0.42	0.37	-0.05
0.85	0.8	-0.05	0.72	0.64	-0.08
0.85	0.8	-0.05	0.72	0.64	-0.08
0.6	0.58	-0.02	0.7	0.64	-0.06
0.6	0.58	-0.02	0.6	0.55	-0.05
0.6	0.58	-0.02	0.6	0.55	-0.05
0.6	0.58	-0.02	0.6	0.55	-0.05
0.53	0.51	-0.02	0.6	0.55	-0.05
0.5	0.51	0.01	0.49	0.45	-0.04
0.5	0.51	0.01	0.49	0.45	-0.04
0.32	0.27	-0.05	0.32	0.3	-0.02
0.22	0.2	-0.02	0.22	0.2	-0.02
0.22	0.2	-0.02	0.22	0.2	-0.02
0.35	0.31	-0.04	0.31	0.22	-0.09
0.35	0.31	-0.04	0.35	0.29	-0.06
0.54	0.51	-0.03	0.35	0.29	-0.06
0.18	0.18	0	0.13	0.13	0
0.18	0.18	0	0.13	0.13	0
0.18	0.18	0	0.11	0.11	0
0.18	0.18	0	0.2	0.2	0
0.18	0.18	0	0.11	0.11	0

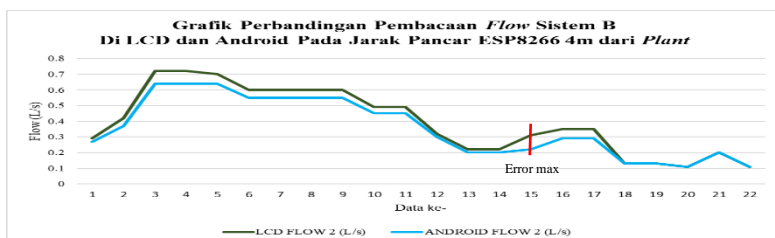
Berdasarkan tabel 4.8 dapat diketahui bahwa pembacaan data *flow* 1 sistem A terdapat error maksimum terbesar 0.06 l/s dan error minimum 0.01 l/s. Sedangkan data *flow* 1 sistem A terdapat error maksimum terbesar 0.09 l/s dan error minimum 0 l/s.

Sehingga didapatkan grafik sebagai berikut.



Gambar 4.8 Grafik Perbandingan Pembacaan *Flow* Sistem A Di LCD dan Android Pada Jarak Pancar ESP8266 4m dari *Plant*

Berdasarkan gambar 4.8 dapat diketahui bahwa pembacaan data *flow* 1 sistem B terdapat error maksimum pada data ke 1 memiliki selisih 0.06 l/s yaitu di LCD terbaca 0.38 l/s sedangkan di android 0.32 l/s. Error minimum pembacaan *flow* 2 pada data ke 10 dan 11 memiliki selisih 0.01 L/s yaitu di LCD terbaca 0.5 l/s sedangkan di android 0.51 l/s.



Gambar 4.9 Grafik Perbandingan Pembacaan *Flow* Sistem B Di LCD dan Android Pada Jarak Pancar ESP8266 4m dari *Plant*

Berdasarkan gambar 4.9 dapat diketahui bahwa sistem A terdapat error maksimum pembacaan *flow* 2 pada data ke 15 memiliki selisih 0.09 L/s yaitu di LCD terbaca 0.31 l/s sedangkan di android 0.22 l/s. Error minimum pembacaan *flow* 1 pada data ke 10 dan 11 memiliki selisih 0.01 L/s yaitu di LCD terbaca 0.5 l/s sedangkan di android 0.51 l/s.

4.2.4 Perbandingan Pembacaan Sensor Pada Jarak Pancar ESP8266 8m dari *Plant*

Tabel 4.9 Data Pembacaan Sistem A Setpoint 25cm Di Android Pada Jarak Pancar ESP8266 8m dari *Plant*

Date	Time	LEVEL 1 (cm)	VOLUME 1 (L)	FLOW 1 (L/s)	TEKANAN 1 (Pa)	MOV (%)
7/23/2018	16:29:56	1.38	0.43	0.58	0.02	90
7/23/2018	16:30:33	4.04	1.27	0.58	0.24	30
7/23/2018	16:30:41	7.79	2.45	0.03	0.23	70
7/23/2018	16:30:45	10.34	3.25	0.03	0.25	30
7/23/2018	16:30:46	18.05	5.67	0.37	0.25	30
7/23/2018	16:30:47	22.26	6.99	0.25	0.23	30
7/23/2018	16:30:49	24.24	7.61	0.42	0.25	30
7/23/2018	16:30:55	25.44	7.99	0.61	0.25	30
7/23/2018	16:31:03	22.91	7.19	0.2	0.25	30
7/23/2018	16:31:04	22.73	7.14	0.12	0.25	30
7/23/2018	16:31:06	23.3	7.32	0.12	0.16	90
7/23/2018	16:31:11	25.02	7.86	0.11	0.25	30
7/23/2018	16:31:22	23.46	7.37	0.13	0.27	30
7/23/2018	16:31:23	22.5	7.07	0.08	0.26	30
7/23/2018	16:31:25	21.39	6.72	0.21	0.26	30
7/23/2018	16:31:32	23.59	7.41	0.16	0.23	70
7/23/2018	16:31:33	24.07	7.56	0.18	0.26	30
7/23/2018	16:31:39	24.89	7.82	0.16	0.25	30
7/23/2018	16:31:41	24.42	7.67	0.16	0.24	30
7/23/2018	16:31:45	24.11	7.57	0.32	0.24	30
7/23/2018	16:31:51	23.26	7.30	0.33	0.22	70
7/23/2018	16:32:00	25.69	8.07	0.2	0.26	30
7/23/2018	16:32:01	25.27	7.93	0.17	0.25	30
7/23/2018	16:32:02	23.84	7.49	0.12	0.24	30
7/23/2018	16:32:04	21.33	6.70	0.31	0.24	30
7/23/2018	16:32:09	22.94	7.20	0.32	0.17	90
7/23/2018	16:32:15	25.27	7.93	0.24	0.25	30
7/23/2018	16:32:20	23.16	7.27	0.33	0.25	30
7/23/2018	16:32:27	21.75	6.83	0.28	0.24	30

Berdasarkan tabel 4.9 dapat diketahui bahwa sistem A dalam mencapai setpoint 25cm terjadi *delay* rata-rata pengiriman data selama 5 detik. Bukan *mov* 30% saat *level* mencapai 25.44 cm dan *flow* 0.25 L/s.

Tabel 4.10 Data Pembacaan Sistem B Setpoint 25cm Di Android Pada Jarak Pancar ESP8266 8m dari Plant

Date	Time	LEVEL 2 (cm)	VOLUME 2 (L)	FLOW 2 (L/s)	TEKANAN 2 (Pa)	PMP (%)
7/23/2018	18:30:58	3.2	1.01	0.63	0.03	90
7/23/2018	18:31:00	5.23	1.64	0.63	0.05	90
7/23/2018	18:31:03	6.01	1.89	0.64	0.06	90
7/23/2018	18:31:09	13.24	4.16	0.64	0.13	90
7/23/2018	18:31:14	16.85	5.29	0.64	0.17	90
7/23/2018	18:31:20	23.26	7.3	0.45	0.23	50
7/23/2018	18:31:25	23.7	7.44	0.18	0.24	50
7/23/2018	18:31:28	25.3	7.94	0.18	0.25	50
7/23/2018	18:31:34	25.78	8.1	0.16	0.26	50
7/23/2018	18:31:39	26.41	8.29	0.15	0.26	50
7/23/2018	18:31:42	20.93	6.57	0.25	0.21	70
7/23/2018	18:31:50	25.25	7.93	0.41	0.25	50
7/23/2018	18:31:56	26.65	8.37	0.15	0.27	50
7/23/2018	18:32:01	22.7	7.13	0.25	0.23	70
7/23/2018	18:32:04	23.68	7.44	0.22	0.24	50
7/23/2018	18:32:09	24.18	7.59	0.16	0.24	50
7/23/2018	18:32:15	23.31	7.32	0.17	0.23	50
7/23/2018	18:32:18	23.42	7.35	0.39	0.23	50
7/23/2018	18:32:20	25.46	8	0.17	0.25	50
7/23/2018	18:32:23	25.93	8.14	0.17	0.26	50
7/23/2018	18:32:26	25.8	8.1	0.16	0.26	50
7/23/2018	18:32:31	24.25	7.62	0.4	0.24	50
7/23/2018	18:32:37	24.64	7.74	0.15	0.25	50
7/23/2018	18:32:43	24.06	7.55	0.23	0.24	50
7/23/2018	18:32:45	25.77	8.09	0.18	0.26	50
7/23/2018	18:32:48	25.11	7.88	0.17	0.25	50
7/23/2018	18:32:54	25.52	8.01	0.18	0.26	50
7/23/2018	18:32:59	23.86	7.49	0.19	0.24	50
7/23/2018	18:33:02	24.06	7.55	0.18	0.24	50

Berdasarkan tabel 4.10 dapat diketahui bahwa sistem B dalam mencapai setpoint 25cm terjadi *delay* rata-rata pengiriman data selama 4 detik. Pompa 50% saat *level* mencapai 25.3 cm, volume 7.94 l, *flow* 0.18 l/s, dan tekanan hidrostatik 0.25 Pa.

Tabel 4.11 Data Pembacaan Sistem A Setpoint 25 cm Di LCD Pada Jarak Pancar ESP8266 8m dari *Plant*

LEVEL 1 (cm)	VOLUME 1 (L)	FLOW 1 (L/s)	TEKANAN 1 (Pa)	MOV (%)
1.65	0.52	0.63	0.01	90
4.14	1.30	0.62	0.04	90
7.85	2.46	0.6	0.07	90
10.39	3.26	0.6	0.09	90
16.69	5.24	0.46	0.15	90
22.35	7.02	0.29	0.20	30
24.27	7.62	0.49	0.22	30
25.64	8.05	0.68	0.23	30
23.06	7.24	0.25	0.21	30
22.92	7.20	0.15	0.21	30
23.45	7.36	0.15	0.21	30
25.21	7.92	0.15	0.23	50
23.68	7.44	0.19	0.21	50
22.79	7.16	0.11	0.21	50
21.46	6.74	0.25	0.19	70
23.77	7.46	0.21	0.21	70
24.27	7.62	0.22	0.22	50
25.05	7.87	0.21	0.23	30
24.64	7.74	0.21	0.22	30
24.84	7.80	0.38	0.22	30
23.88	7.50	0.38	0.21	50
26.37	8.28	0.28	0.24	30
25.55	8.02	0.23	0.23	30
23.11	7.26	0.15	0.21	50
21.56	6.77	0.35	0.19	70
23.87	7.50	0.36	0.21	70
25.32	7.95	0.27	0.23	30
23.66	7.43	0.37	0.21	70
21.94	6.89	0.34	0.20	70

Berdasarkan tabel 4.11 dapat diketahui bahwa sistem A dalam mencapai setpoint 25cm terjadi *delay* rata-rata pengiriman data selama 5 detik. Bukan mov 30% saat *level* mencapai 25.64 cm dan *flow* 0.68 l/s.

**Tabel 4.12 Data Pembacaan Sistem B Setpoint 25 cm Di LCD
Pada Jarak Pancar ESP8266 8m dari *Plant***

LEVEL 2 (cm)	VOLUME 2 (L)	FLOW 2 (L/s)	TEKANAN 2 (Pa)	PMP (%)
3.41	1.07	0.7	0.03	90
5.44	1.71	0.7	0.05	90
6.24	1.96	0.67	0.06	90
13.51	4.24	0.67	0.12	90
16.96	5.33	0.65	0.15	90
23.49	7.38	0.47	0.21	50
23.86	7.49	0.23	0.21	50
25.56	8.03	0.21	0.23	50
25.98	8.16	0.19	0.23	50
26.73	8.39	0.19	0.24	50
21.14	6.64	0.27	0.19	70
25.33	7.95	0.42	0.23	50
26.81	8.42	0.16	0.24	50
22.85	7.17	0.27	0.21	70
23.99	7.53	0.24	0.22	50
24.32	7.64	0.19	0.22	50
23.66	7.43	0.2	0.21	50
23.76	7.46	0.41	0.21	50
25.61	8.04	0.2	0.23	50
26.14	8.21	0.21	0.24	50
25.99	8.16	0.18	0.23	50
25.74	8.08	0.47	0.23	50
24.82	7.79	0.21	0.22	50
24.47	7.68	0.29	0.22	50
25.83	8.11	0.24	0.23	50
25.22	7.92	0.22	0.23	50
25.47	8.00	0.21	0.23	50
24.01	7.54	0.23	0.22	50
24.23	7.61	0.22	0.22	50

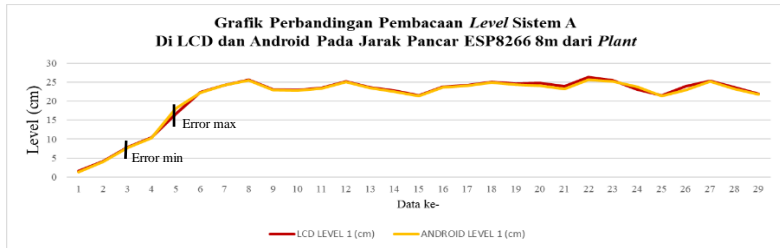
Berdasarkan tabel 4.12 dapat diketahui bahwa sistem B dalam mencapai setpoint 25cm terjadi *delay* rata-rata pengiriman data selama 4 detik. Tegangan pompa 50% saat *level* mencapai 25.56 cm dan *flow* 0.21 l/s.

LCD	ANDROID	
LEVEL 1 (cm)	LEVEL 1 (cm)	Error
1.65	1.38	-0.27
4.14	4.04	-0.1
7.85	7.79	-0.06
10.39	10.34	-0.05
16.69	18.05	1.36
22.35	22.26	-0.09
24.27	24.24	-0.03
25.64	25.44	-0.2
23.06	22.91	-0.15
22.92	22.73	-0.19
23.45	23.3	-0.15
25.21	25.02	-0.19
23.68	23.46	-0.22
22.79	22.5	-0.29
21.46	21.39	-0.07
23.77	23.59	-0.18
24.27	24.07	-0.2
25.05	24.89	-0.16
24.64	24.42	-0.22
24.84	24.11	-0.73
23.88	23.26	-0.62
26.37	25.69	-0.68
25.55	25.27	-0.28
23.11	23.84	0.73
21.56	21.33	-0.23
23.87	22.94	-0.93
25.32	25.27	-0.05
23.66	23.16	-0.5
21.94	21.75	-0.19

LCD	ANDROID	
LEVEL 2 (cm)	LEVEL 2 (cm)	Error
3.41	3.2	-0.21
5.44	5.23	-0.21
6.24	6.01	-0.23
13.51	13.24	-0.27
16.96	16.85	-0.11
23.49	23.26	-0.23
23.86	23.7	-0.16
25.56	25.3	-0.26
25.98	25.78	-0.2
26.73	26.41	-0.32
21.14	20.93	-0.21
25.33	25.25	-0.08
26.81	26.65	-0.16
22.85	22.7	-0.15
23.99	23.68	-0.31
24.32	24.18	-0.14
23.66	23.31	-0.35
23.76	23.42	-0.34
25.61	25.46	-0.15
26.14	25.93	-0.21
25.99	25.8	-0.19
25.74	24.25	-1.49
24.82	24.64	-0.18
24.47	24.06	-0.41
25.83	25.77	-0.06
25.22	25.11	-0.11
25.47	25.52	0.05
24.01	23.86	-0.15
24.23	24.06	-0.17

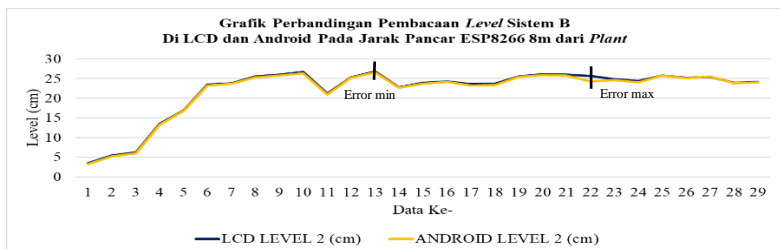
Berdasarkan tabel 4.13 dapat diketahui bahwa pembacaan data *level 1* sistem A terdapat error maksimum terbesar 1.36 cm dan error minimum 0.06 cm. Sedangkan data *level 2* sistem B terdapat error maksimum terbesar 1.49 cm dan error minimum 0.05 cm.

Sehingga didapatkan grafik sebagai berikut.



Gambar 4.10 Grafik Perbandingan Pembacaan *Level* Sistem A Di LCD dan Android Pada Jarak Pancar ESP8266 8m dari *Plant*

Berdasarkan gambar 4.6 dapat diketahui bahwa pembacaan data *level* 1 sistem A terdapat error maksimum pada data ke 5 memiliki selisih 1.36 cm yaitu di LCD terbaca 16.69 cm sedangkan di android 18.05 cm. Error minimum data ke 3 memiliki selisih 0.06 cm yaitu di LCD terbaca 7.85 cm sedangkan di android 7.79 cm.



Gambar 4.11 Grafik Perbandingan Pembacaan *Level* Sistem B Di LCD dan Android Pada Jarak Pancar ESP8266 8m dari *Plant*

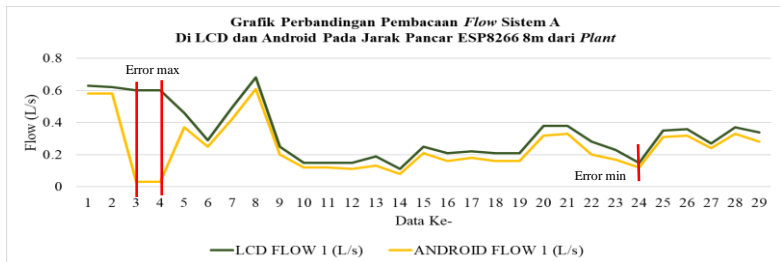
Berdasarkan gambar 4.7 dapat diketahui bahwa pembacaan data *level* 2 sistem B terdapat error maksimum pada data ke 22 memiliki selisih 1.49 cm yaitu di LCD terbaca 25.74 cm sedangkan di android 15.53 cm. Error minimum data ke 13 memiliki selisih 0.09 cm yaitu di LCD terbaca 24.88 cm sedangkan di android 24.25 cm.

Tabel 4.14 Perbandingan Data Pembacaan *Flow* Setpoint 25 cm LCD dan Android Pada Jarak Pancar ESP8266 8m dari *Plant*

LCD	ANDROID		LCD	ANDROID	
FLOW 1 (L/s)	FLOW 1 (L/s)	Error	FLOW 2 (L/s)	FLOW 2 (L/s)	Error
0.63	0.58	-0.05	0.7	0.63	-0.07
0.62	0.58	-0.04	0.7	0.63	-0.07
0.6	0.03	-0.57	0.67	0.64	-0.03
0.6	0.03	-0.57	0.67	0.64	-0.03
0.46	0.37	-0.09	0.65	0.64	-0.01
0.29	0.25	-0.04	0.47	0.45	-0.02
0.49	0.42	-0.07	0.23	0.18	-0.05
0.68	0.61	-0.07	0.21	0.18	-0.03
0.25	0.2	-0.05	0.19	0.16	-0.03
0.15	0.12	-0.03	0.19	0.15	-0.04
0.15	0.12	-0.03	0.27	0.25	-0.02
0.15	0.11	-0.04	0.42	0.41	-0.01
0.19	0.13	-0.06	0.16	0.15	-0.01
0.11	0.08	-0.03	0.27	0.25	-0.02
0.25	0.21	-0.04	0.24	0.22	-0.02
0.21	0.16	-0.05	0.19	0.16	-0.03
0.22	0.18	-0.04	0.2	0.17	-0.03
0.21	0.16	-0.05	0.41	0.39	-0.02
0.21	0.16	-0.05	0.2	0.17	-0.03
0.38	0.32	-0.06	0.21	0.17	-0.04
0.38	0.33	-0.05	0.18	0.16	-0.02
0.28	0.2	-0.08	0.47	0.4	-0.07
0.23	0.17	-0.06	0.21	0.15	-0.06
0.15	0.12	-0.03	0.29	0.23	-0.06
0.35	0.31	-0.04	0.24	0.18	-0.06
0.36	0.32	-0.04	0.22	0.17	-0.05
0.27	0.24	-0.03	0.21	0.18	-0.03
0.37	0.33	-0.04	0.23	0.19	-0.04
0.34	0.28	-0.06	0.22	0.18	-0.04

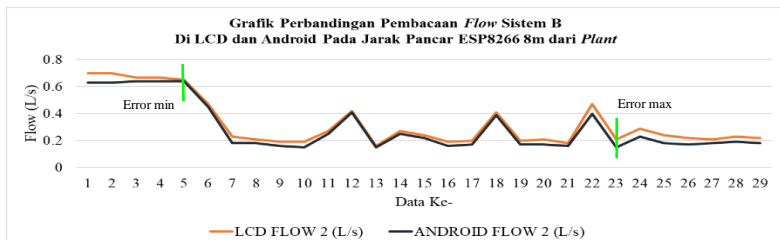
Berdasarkan tabel 4.14 dapat diketahui bahwa pembacaan data *flow* 1 sistem A terdapat error maksimum terbesar 0.57 l/s dan error minimum 0.03 l/s. Sedangkan data *flow* 2 sistem B terdapat error maksimum terbesar 0.07 l/s dan error minimum 0.01 l/s.

Sehingga didapatkan grafik sebagai berikut.



Gambar 4.12 Grafik Perbandingan Pembacaan Flow Sistem A Di LCD dan Android Pada Jarak Pancar ESP8266 8m dari Plant

Berdasarkan gambar 4.8 dapat diketahui bahwa pembacaan data *flow* 1 sistem A terdapat error maksimum pada data ke 3 dan 4 memiliki selisih 0.57 l/s yaitu di LCD terbaca 0.6 cm sedangkan di android 0.03 l/s. Error minimum data ke 24 memiliki selisih 0.03 cm yaitu di LCD terbaca 0.15 l/s sedangkan di android 0.12 l/s.



Gambar 4.13 Grafik Perbandingan Pembacaan Flow Sistem B Di LCD dan Android Pada Jarak Pancar ESP8266 8m dari Plant

Berdasarkan grafik 4.9 dapat diketahui bahwa pembacaan data *flow* 1 sistem A terdapat error maksimum pada data ke 23 memiliki selisih 0.07 l/s yaitu di LCD terbaca 0.47 l/s sedangkan di android 0.4 l/s. Error minimum data ke 5 memiliki selisih 0.01 Ls yaitu di LCD terbaca 0.65 L/s sedangkan di android 0.64 l/s.

4.2.5 Perbandingan Pembacaan Sensor Pada Jarak Pancar ESP8266 10m dari *Plant*

Tabel 4.15 Data Pembacaan Sistem A Setpoint 25cm Di Android Pada Jarak Pancar ESP8266 10m dari *Plant*

Date	Time	LEVEL 1 (cm)	VOLUME 1 (L)	FLOW 1 (L/s)	TEKANAN 1 (Pa)	MOV (%)
7/23/2018	17:09:26	1.78	0.56	0	0.05	80
7/23/2018	17:09:31	1.62	0.51	0	0.05	80
7/23/2018	17:09:34	3.64	1.14	0	0.10	80
7/23/2018	17:09:40	8.92	2.80	0	0.25	80
7/23/2018	17:09:45	12.25	3.85	0	0.35	70
7/23/2018	17:09:51	17.77	5.58	0	0.50	70
7/23/2018	17:09:56	24.03	7.55	0	0.68	35
7/23/2018	17:10:02	25.02	7.86	0	0.71	35
7/23/2018	17:10:04	19.69	6.18	0	0.56	70
7/23/2018	17:11:13	19.43	6.10	0	0.55	80
7/23/2018	17:11:15	22.86	7.18	0	0.65	75
7/23/2018	17:11:18	24.89	7.82	0	0.70	35
7/23/2018	17:11:21	22	6.91	0	0.62	70
7/23/2018	17:11:24	17.6	5.53	0	0.50	80

Berdasarkan tabel 4.15 dapat diketahui bahwa sistem A dalam mencapai setpoint 25cm terjadi *delay* rata-rata pengiriman data selama 9 detik. Bukan mov 35% saat *level* mencapai 25.02 cm dan *flow* 0 l/s atau tidak terbaca.

Tabel 4.16 Data Pembacaan Sistem B Setpoint 25cm Di Android Pada Jarak Pancar ESP8266 10m dari *Plant*

Date	Time	LEVEL 2 (cm)	VOLUME 2 (L)	FLOW 2 (L/s)	TEKANAN 2 (Pa)	PMP (%)
7/23/2018	17:09:26	2.57	0.81	0	0.07	90
7/23/2018	17:09:31	5.59	1.76	0	0.16	90
7/23/2018	17:09:34	9.12	2.86	0	0.26	60
7/23/2018	17:09:40	12.01	3.77	0	0.34	60
7/23/2018	17:09:45	19.57	6.14	0	0.55	45
7/23/2018	17:09:51	24.48	7.69	0	0.69	30
7/23/2018	17:09:56	25.65	8.05	0	0.72	30
7/23/2018	17:10:02	22.22	6.98	0	0.63	50
7/23/2018	17:10:04	18.28	5.74	0	0.52	50
7/23/2018	17:11:13	20.45	6.42	0	0.58	70
7/23/2018	17:11:15	23.59	7.41	0	0.67	60
7/23/2018	17:11:18	24.56	7.71	0	0.69	45
7/23/2018	17:11:21	25.54	8.02	0	0.72	30
7/23/2018	17:11:24	21.67	6.80	0	0.61	50

Berdasarkan tabel 4.16 dapat diketahui bahwa sistem B dalam mencapai setpoint 25cm terjadi *delay* rata-rata pengiriman data selama 9 detik. Tegangan pompa 30% saat *level* mencapai 25.65 cm dan *flow* 0 l/s atau tidak terbaca.

Tabel 4.17 Data Pembacaan Sistem A Setpoint 25cm Di LCD Pada Jarak Pancar ESP8266 10m dari Plant

LEVEL 1 (cm)	VOLUME 1 (L)	FLOW 1 (L/s)	TEKANAN 1 (p)	MOV (%)
2.1	0.66	0.38	0.02	90
2.2	0.69	0.5	0.02	90
2.77	0.87	0.85	0.03	90
4	1.26	0.85	0.04	90
6.5	2.04	0.6	0.07	90
10.11	3.17	0.6	0.1	90
13	4.08	0.6	0.13	90
15.37	4.83	0.6	0.16	90
15.43	4.85	0.53	0.16	90
18.71	5.87	0.5	0.19	90
23.89	7.50	0.5	0.24	89
25.08	7.88	0.32	0.25	89
24.88	7.81	0.22	0.25	89
24.19	7.60	0.22	0.24	89
21.48	6.74	0.35	0.22	70
25.55	8.02	0.35	0.26	89
25.33	7.95	0.54	0.25	89
25.7	8.07	0.18	0.26	89
25.4	7.98	0.18	0.25	89
25.4	7.98	0.18	0.25	89
25.3	7.94	0.18	0.26	89
24.9	7.82	0.18	0.25	89

Berdasarkan tabel 4.17 dapat diketahui bahwa sistem A dalam mencapai setpoint 25cm terjadi *delay* rata-rata pengiriman data selama 9 detik. Bukaan MOV 89% saat *level* mencapai 25.08 cm dan *flow* 0.21 l/s.

**Tabel 4.18 Data Pembacaan Sistem B Setpoint 25cm Di LCD
Pada Jarak Pancar ESP8266 10m dari *Plant***

LEVEL 2 (cm)	VOLUME 2 (L)	FLOW 2 (L/s)	TEKANA N 2 (p)	PMP (%)
3.12	0.98	0.29	0.03	90
2.62	0.82	0.42	0.03	90
3.36	1.06	0.72	0.04	90
5.6	1.76	0.72	0.06	90
9.08	2.85	0.7	0.09	90
12.41	3.90	0.6	0.13	90
14.08	4.42	0.6	0.14	90
15.38	4.83	0.6	0.15	90
19.32	6.07	0.6	0.19	90
21.34	6.70	0.49	0.21	70
20.7	6.50	0.49	0.21	70
24.43	7.67	0.32	0.25	50
25.32	7.95	0.22	0.25	50
24.06	7.55	0.22	0.24	50
23.9	7.50	0.31	0.25	50
22.45	7.05	0.35	0.23	70
23.37	7.34	0.35	0.23	50
24.48	7.69	0.13	0.25	50
23.13	7.26	0.13	0.23	50
25.21	7.92	0.11	0.25	50
23.8	7.47	0.2	0.24	50
24.58	7.72	0.11	0.25	50

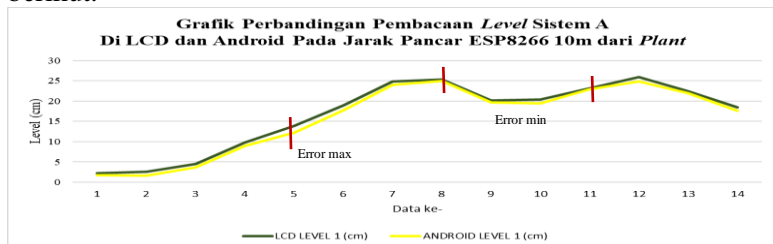
Berdasarkan tabel 4.18 dapat diketahui bahwa sistem A dalam mencapai setpoint 25cm terjadi *delay* rata-rata pengiriman data selama 9 detik. Tegangan pompa 50% saat *level* mencapai 25.32 cm dan *flow* 0.25 l/s.

Tabel 4.19 Perbandingan Data Pembacaan *Level* Setpoint 25 cm LCD dan Android Pada Jarak Pancar ESP8266 10m dari *Plant*

LCD	ANDROID		LCD	ANDROID	
LEVEL 1 (cm)	LEVEL 1 (cm)	Error	LEVEL 2 (cm)	LEVEL 2 (cm)	Error
2.27	1.78	-0.49	2.8	2.17	-0.63
2.55	1.62	-0.93	5.73	5	-0.73
4.53	3.64	-0.89	9.36	9	-0.36
9.77	8.92	-0.85	12.21	11.21	-1
13.91	12.25	-1.66	19.74	17.27	-2.47
18.96	17.77	-1.19	24.38	23.01	-1.37
24.87	24.03	-0.84	25.85	25.05	-0.8
25.32	25.02	-0.3	22.01	21.52	-0.49
20.2	19.69	-0.51	18.18	17.4	-0.78
20.39	19.43	-0.96	20.69	20.15	-0.54
23.16	22.86	-0.3	23.85	21.79	-2.06
25.93	24.89	-1.04	24.77	24.36	-0.41
22.41	22	-0.41	25.9	25.4	-0.5
18.5	17.6	-0.9	21.87	21.27	-0.6

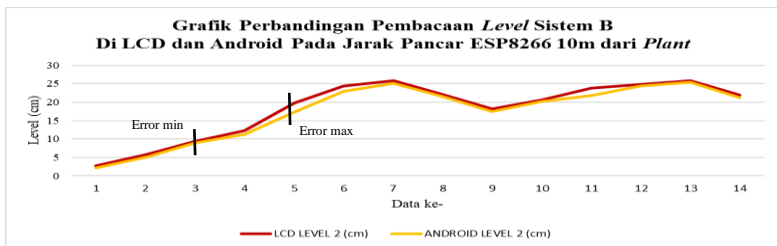
Berdasarkan tabel 4.19 dapat diketahui bahwa pembacaan data *level 1* sistem A terdapat error maksimum terbesar 1.66 cm dan error minimum 0.3 cm. Sedangkan data *level 2* sistem B terdapat error maksimum terbesar 2.47 cm dan error minimum 0.36 cm.

Sehingga dari data tersebut didapatkan grafik sebagai berikut.



Gambar 4.14 Grafik Perbandingan Pembacaan *Level* Sistem A Di LCD dan Android Pada Jarak Pancar ESP8266 10m dari *Plant*

Berdasarkan gambar 4.10 dapat diketahui bahwa pembacaan data *level* 1 sistem A terdapat error maksimum pada data ke 5 memiliki selisih 1.66 cm yaitu di LCD terbaca 13.91 cm sedangkan di android 12.25 cm. Error minimum data ke 8 dan 12 memiliki selisih 0.3 cm yaitu di LCD terbaca 25.32 cm sedangkan di android 25.02 untuk data ke 8 dan di LCD terbaca 2.16 cm sedangkan di android 22.86 untuk data ke 12.



Gambar 4.15 Grafik Perbandingan Pembacaan *Level* Sistem A Di LCD dan Android Pada Jarak Pancar ESP8266 10m dari *Plant*

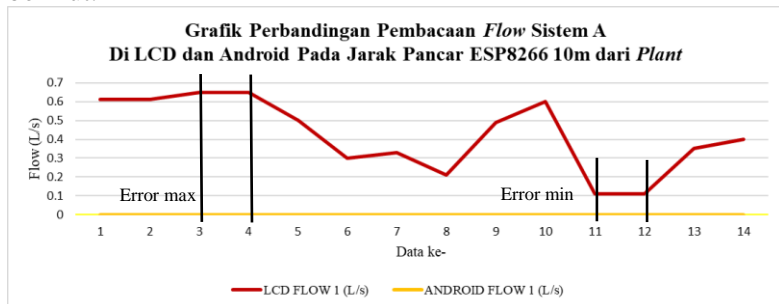
Berdasarkan gambar 4.11 dapat diketahui bahwa pembacaan data *level* 2 sistem B terdapat error maksimum pada data ke 3 dan 4 memiliki selisih 2.47 cm yaitu di LCD terbaca 19.74 cm sedangkan di android 17.27 cm. Error minimum data ke 3 memiliki selisih 0.36 cm yaitu di LCD terbaca 9.36 cm sedangkan di android 9 cm.

Tabel 4.20 Perbandingan Data Pembacaan *Flow* Setpoint 25 cm LCD dan Android Pada Jarak Pancar ESP8266 10m dari Plant

LCD	ANDROID		LCD	ANDROID	
FLOW 1 (L/s)	FLOW 1 (L/s)	Error	FLOW 2 (L/s)	FLOW 2 (L/s)	Error
0.61	0	-0.61	0.56	0	-0.56
0.61	0	-0.61	0.56	0	-0.56
0.65	0	-0.65	0.58	0	-0.58
0.65	0	-0.65	0.58	0	-0.58
0.5	0	-0.5	0.54	0	-0.54
0.3	0	-0.3	0.22	0	-0.22
0.33	0	-0.33	0.19	0	-0.19
0.21	0	-0.21	0.21	0	-0.21
0.49	0	-0.49	0.8	0	-0.8
0.6	0	-0.6	0.49	0	-0.49
0.11	0	-0.11	0.49	0	-0.49
0.11	0	-0.11	0.18	0	-0.18
0.35	0	-0.35	0.3	0	-0.3
0.4	0	-0.4	0.41	0	-0.41

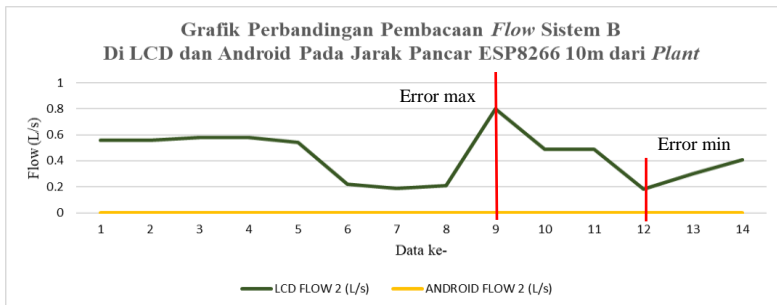
Berdasarkan tabel 4.20 dapat diketahui bahwa pembacaan data *flow* 1 sistem A terdapat error maksimum terbesar 0.65 l/s dan error minimum 0.11 l/s. Sedangkan data *flow* 2 sistem B terdapat error maksimum terbesar 0.8 l/s dan error minimum 0.18 l/s.

Sehingga dari data tersebut didapatkan grafik sebagai berikut.



Gambar 4.16 Grafik Perbandingan Pembacaan *Flow* Sistem A Di LCD dan Android Pada Jarak Pancar ESP8266 10m dari Plant

Berdasarkan gambar 4.12 dapat diketahui bahwa pembacaan data *flow* 1 sistem A terdapat error maksimum pada data ke 3 dan 4 memiliki selisih 0.65 l/s yaitu di LCD terbaca 0.65 l/s sedangkan di android 0 l/s atau tidak terbaca. Error minimum pada data ke 11 dan 12 memiliki selisih 0.11 l/s yaitu di LCD terbaca 0.11 l/s sedangkan di android 0 l/s atau tidak terbaca.



Gambar 4.17 Grafik Perbandingan Pembacaan Flow Sistem B Di LCD dan Android Pada Jarak Pancar ESP8266 10m dari Plant

Berdasarkan gambar 4.13 dapat diketahui bahwa pembacaan data *flow* 2 sistem B terdapat error maksimum pada data ke 9 memiliki selisih 0.8 l/s yaitu di LCD terbaca 0.8 l/s sedangkan di android 0 l/s atau tidak terbaca. Error minimum pada data ke 12 memiliki selisih 0.18 l/s yaitu di LCD terbaca 0.18 l/s sedangkan di android 0 l/s atau tidak terbaca.

4.3 Pembahasan

Pada tugas akhir ini digunakan modul wifi ESP8266 untuk mentransmisikan data dari sensor ultrasonik HC-SR04 dan *water flow meter* ke *database MySQL web server*. Dimana ESP8266 mendapatkan input tegangan VCC dari PORTD5 mikrokontroler AVR Atmega 1284P. Karena adanya perbedaan nilai tegangan input VCC ESP8266 3.3 volt dan VCC AVR Atmega 1284P 5 volt

maka diperlukan adanya regulator untuk menurunkan tegangan Atmega 1284P agar data dapat ditransmisikan dengan baik.

Komunikasi serial dari modul wifi ESP8266 ke Atmega 1284P menggunakan UART (TX) dan (RX) yaitu (TX) ESP8266 terhubung dengan (RX) Atmega 1284P. Komunikasi serial dilakukan dengan menginisialisasi pin (TX) 12 byte dan (RX) 14 byte, inisialisasi *server* <https://www.myhyts.000webhost.com/>, inisialisasi *serial baud rate* 9600 Hz, inisialisasi *client* agar terhubung dengan ESP8266 “ESP_FD8818” yang dihubungkan ke hotspot portable HP dengan SSID “Prime7” dan password “riha0895”, membaca alamat HTTP, mengirim serial ke Atmega 1284P, membaca data dari Atmega 1284P berbentuk digital diubah dalam bentuk string agar dapat dikirimkan ke database web server, membuat string HTTP hingga dapat menampilkan data. Pada program Atmega 1284P perintah komunikasi serial ESP8266 berupa `printf (“alamat http upload menu database”+string(variable sesuai database) dan printf(“.php”)`.

Data yang dikirim ke database MySQL ditampilkan pada aplikasi android pada menu *database* dimana database dapat diekspor dalam bentuk tabel excel (.xls), *monitoring* yang berisi status *plant* saat dijalankan dan *history* berisi riwayat *plant* dijalankan.

Dari pengambilan data dengan 3 jarak yang berbeda didapatkan hasil pembacaan pada jarak pancar ESP8266 4 m dari *plant*, error maksimum *level* 1 0.41 cm dan *level* 2 1.15 cm dengan *delay* pengiriman data 3 detik, error maksimum *flow* 1 0.06 l/s dan *flow* 2 0.09 l/s dengan *delay* pengiriman data 3 detik. 8 m dari *plant*, error maksimum *level* 1 1.36 cm dan *level* 2 1.49 cm dengan *delay* pengiriman data 5 detik, *flow* 1 0.57 l/s dan *flow* 2 0.07 l/s dengan *delay* pengiriman data 4 detik. 10 m dari *plant*, error maksimum *level* 1 1.66 cm dan *level* 2 2.47 cm dengan *delay* pengiriman data 9 detik, *flow* 1 0.65 l/s dan *flow* 2 0.8 l/s dengan *delay* pengiriman data 9 detik. Sehingga dapat disimpulkan bahwa semakin jauh jarak pancar ESP8266 dengan *plant* maka semakin besar nilai error maksimum dari data tersebut.

Halaman ini memang dikosongkan

Bab V

Penutup

5.1 Kesimpulan

Adapun kesimpulan dari pembuatan rancang bangun sistem *monitoring* jarak jauh pada *hydrostatic training system plant* skala laboratorium ini adalah sebagai berikut.

- 1) Telah dibuat sistem *monitoring* jarak jauh pada *hydrostatic training system plant* skala laboratorium untuk *variable level* dan *flow* menggunakan modul Wifi ESP8266 sebagai *transmitter* data dari sensor ke *database MySQL web server* yang ditampilkan pada aplikasi android.
- 2) Dari pengambilan data dengan 3 jarak yang berbeda didapatkan hasil pembacaan pada jarak pancar ESP8266 4 m dari *plant*, error maksimum *level* 1 0.41 cm dan *level* 2 1.15 cm dengan *delay* pengiriman data 3 detik, error maksimum *flow* 1 0.06 l/s dan *flow* 2 0.09 l/s dengan *delay* pengiriman data 3 detik. 8 m dari *plant*, error maksimum *level* 1 1.36 cm dan *level* 2 1.49 cm dengan *delay* pengiriman data 5 detik, *flow* 1 0.57 L/s dan *flow* 2 0.07 l/s dengan *delay* pengiriman data 4 detik. 10 m dari *plant*, error maksimum *level* 1 1.66 cm dan *level* 2 2.47 cm dengan *delay* pengiriman data 9 detik, *flow* 1 0.65 l/s dan *flow* 2 0.8 l/s dengan *delay* pengiriman data 9 detik.
- 3) Sehingga dapat disimpulkan bahwa jarak pengiriman data ESP8266 dari *plant* berbanding lurus dengan error maksimum data dan *delay* pengiriman data.

5.2 Saran

Adapun saran untuk pengembangan penelitian tugas akhir rancang bangun sistem *mmonitoring* jarak jauh pada *hydrostatic training system plant* skala laboratorium ini adalah dibuatkan menu *setpoint* yang dapat diubah dari aplikasi dan HMI dengan progress bar yang dapat berjalan seperti animasi.

Halaman ini memang dikosongkan

Daftar Pustaka

- [1] M. A. Crabtree, "Industrial Flow Measurement," University of Huddersfield, 2009.
- [2] R. H. Perry, D. W. Green, and J. O. Maloney, *Chemical Engineers ' Handbook Seventh*, Seventh., vol. 27. New York: McGraw-Hill, 1997.
- [3] T. Al-Shemmeri, *Engineering Fluid Mechanics*. Ventus Publishing ApS, 2012.
- [4] Mercy Corps, "Design, Monitoring and Evaluation Guidebook," 2005.
- [5] J. P. Bentley, *Principles of Measurement Systems*, vol. 30, no. 1. 2005.
- [6] E. J. Morgan, "HC - SR04 Ultrasonic Sensor," 2014.
- [7] C. Technologies, "User ' s Manual HC-SR04 Ultrasonic Sensor," no. May, pp. 1–10, 2013.
- [8] Honeywell, "Hall Effect Sensing and Application," *Sens. Control*, p. 126, 2011.
- [9] Openhacks, "G1" Water Flow Sensor."
- [10] Atmel, *ATmega1284P*. San Jose, USA: Atmel Corporation, 2009.
- [11] Synacorp, "NODEMCU Lua IoT ESP8266 Wifi Controller Board V3."
- [12] M. Kashyap, V. Sharma, and N. Gupta, "Taking MQTT and NodeMcu to IOT: Communication in Internet of Things," in *Procedia Computer Science*, 2018, vol. 132, no. Iccids, pp. 1611–1618.
- [13] C. J. Date, *An Introduction to Database Systems*. Pearson Education, Inc, 2004.
- [14] J. Greenspan and B. Bulger, *MySQL/PHP database applications*. Foster City, CA: M&T Books, 2001.
- [15] M. Mazlakowski and T. Butcher, *Teach Yourself MySQL in 21 Days*. Sams, 2000.
- [16] W. M. Lee, *Beginning Android™ Application Development*. Indianapolis, Indiana: Wiley Publishing,

Inc, 2011.

- [17] J. F. DiMarzio, *Android Programing with Android Studio*. Indianapolis, Indiana: John Wiley & Sons, Inc, 2017.
- [18] A. ATC, *Java Fundamentals for AndroidTM Development*. Android ATC, 2017.

Lampiran A

Listing Program AVR Atmega 1284P

```
#include <mega1284p.h>
#include <alcd.h>
#include <delay.h>
#include <stdio.h>
#include <stdlib.h>

#define putus 1
#define konek 0

#define nyala 1
#define mati 0

#define dn PINB.0
#define ok PINB.1
#define up PINB.2

#define trb PORTC.2
#define ecb PINC.1

#define tra PORTC.3
#define eca PINC.4

#define vala PORTC.7
#define valb PORTC.6
#define buz PORTC.5
#define inda PORTB.6
#define indb PORTB.7

eeprom float saveFlowKa[100];
eeprom float saveFlowKi[100];
eeprom float saveLevelKa[100];
eeprom float saveLevelKi[100];
eeprom int saveSplvl;
eeprom int savePeriode;
eeprom int saveMov;
eeprom int savePmp;

float flowka,flowki,leveka,levelki;
int counterka,counterki;
```

```

int cntlvlka,cntlvlki;
int time,jaraka,jarakaCm;
float levelka;
int jaraki,jarakiCm;
float levelki;
float vola,voli,presa,presi;
int msdetik,i,mode,moni;
unsigned int data,data2;
int mov,pmp,periode,splvl;
char
buff[16],tempflowka[10],tempflowki[10],templevelka[10],templevelki[10],temp
pres[10],tempvol[10];
int mili,detik;

```

```

void bacalevelka();
void bacalevelki();
void setpin();
void setMov(unsigned int persen);
void setPmp(unsigned int persena);
void tespmp();
void tesmov();
void aktifmov();
void nonaktifmov();
void aktifpmp();
void nonaktifpmp();
void menu();
void setting();
void setWifi();
void kontrol();

```

```

interrupt [EXT_INT0] void ext_int0_isr(void)
{
    counterki++;
}

```

```

interrupt [EXT_INT1] void ext_int1_isr(void)
{
    counterka++;
}

```

```

interrupt [TIM0_OVF] void timer0_ovf_isr(void)
{
    TCNT0=0x64;
}

```

```

mili++;
if(mili>10){
    flowka = (float) counterka*0.1/4.8;
    flowki = (float) counterki*0.1/4.8;

    bacalevelka();
    bacalevelki();

    vola   = (float) 3.14*0.01*levelka*10;
    presa  = (float) leveka/100;
    voli   = (float) 3.14*0.01*levelki*10;
    presi  = (float) levelki/100;

    kontrol();

    mili=0;
    msdetik++;
    if(msdetik>=10){
        detik++;
        i++;
        if(detik>=periode){
            flowka=saveFlowKa[i];
            flowki=saveFlowKi[i];
            levelka=saveLevelKa[i];
            levelki=saveLevelKi[i];
            detik=0;

//set esp to transmit data (Andiah A. 10511400000052)
printf("upmoni.php?lvla="+String(levela)+"&vola="+String(vola)+"&floa="+String(flowa)+"&prea="+String(presa)+"&lvlb="+String(levelb)+"&volb="+String(volb)+"&flob="+String(flowb)+"&preb="+String(presb)+"&mov="+String(mov)+"&mtr="+String(mtr));
        }
    }

    if(!up){moni++; delay_ms(50);}
    if(!dn){moni--; delay_ms(50);}
    if(!ok){
        delay_ms(50);
        lcd_clear();
        lcd_putsf("Saving EEPROM...");
        for(i=0;i<20;i++){
            lcd_gotoxy(i,1);

```

```

    lcd_putchar(0xff);
    delay_ms(100);
}
lcd_clear();
for(;;){
    lcd_gotoxy(0,0);
    lcd_putsf("Save Ok");
    lcd_gotoxy(0,1);
    lcd_putsf("Tekan tombol hitam");
}
}
if(moni<1) {moni=4;}
if(moni>4) {moni=1;}

if(moni==1){
    lcd_gotoxy(8,0); lcd_putsf("Kiri ");
    lcd_gotoxy(0,1); lcd_putsf("Mov :    % ");
    lcd_gotoxy(6,1); sprintf(buff,"%3d",mov); lcd_puts(buff);
    lcd_gotoxy(0,2); lcd_putsf("Flw :    L/s");
    lcd_gotoxy(6,2); ftoa(flowki,2,tempflowki); lcd_puts(tempflowki);
    lcd_gotoxy(0,3); lcd_putsf("Lvl :    cm ");
    lcd_gotoxy(6,3); ftoa(levelki,2,templevelki); lcd_puts(templevelki);
}

else if(moni==2){
    lcd_gotoxy(8,0); lcd_putsf("Kiri ");
    lcd_gotoxy(0,1); lcd_putsf("Prs :    kpa");
    lcd_gotoxy(6,1); ftoa(presi,2,temppres); lcd_puts(temppres);
    lcd_gotoxy(0,2); lcd_putsf("Vol :    L ");
    lcd_gotoxy(6,2); ftoa(voli,2,tempvol); lcd_puts(tempvol);
    lcd_gotoxy(0,3); lcd_putsf("    ");
}

else if(moni==3){
    lcd_gotoxy(8,0); lcd_putsf("Kanan");
    lcd_gotoxy(0,1); lcd_putsf("Pmp :    % ");
    lcd_gotoxy(6,1); sprintf(buff,"%3d",pmp); lcd_puts(buff);
    lcd_gotoxy(0,2); lcd_putsf("Flw :    L/s");
    lcd_gotoxy(6,2); ftoa(flowka,2,tempflowka); lcd_puts(tempflowka);
    lcd_gotoxy(0,3); lcd_putsf("Lvl :    cm ");
    lcd_gotoxy(6,3); ftoa(levelka,2,templevelka); lcd_puts(templevelka);
}
}

```



```

else if(moni==4){
    lcd_gotoxy(8,0); lcd_putsf("Kanan");
    lcd_gotoxy(0,1); lcd_putsf("Prs :      kpa");
    lcd_gotoxy(6,1); ftoa(presa,2,temppres); lcd_puts(temppres);
    lcd_gotoxy(0,2); lcd_putsf("Vol :      L ");
    lcd_gotoxy(6,2); ftoa(vola,2,tempvol); lcd_puts(tempvol);
    lcd_gotoxy(0,3); lcd_putsf("                ");
}
}
}

```

```

void setpin(){
    DDRB.0=0; PORTB.0=1; // dn
    DDRB.1=0; PORTB.1=1; // ok
    DDRB.2=0; PORTB.1=1; // up

    DDRD.2=0; PORTD.2=1; // Interrupt Flowkanan
    DDRD.3=0; PORTD.3=1; // Interrupt Flowkiri

    DDRC.2=1; PORTC.2=0; // Trigger level kiri
    DDRC.1=0; PORTC.1=1; // Echo level kiri

    DDRC.3=1; PORTC.3=0; // Trigger level kanan
    DDRC.4=0; PORTC.4=1; // Echo level kanan

    DDRC.7=1; PORTC.7=putus; // Valve kanan
    DDRC.6=1; PORTC.6=putus; // Valve kiri
    DDRC.5=1; PORTC.5=putus; // Bel

    DDRB.7=1; PORTB.7=mati; // indikator kanan
    DDRB.6=1; PORTB.6=mati; // indikator kiri
}

```

```

void bacalevelka(){
    tra=1;
    delay_us(10);
    tra=0;

    for(;;){
        if(eca==1){
            cntlvlka=0;
            break;
        }
    }
}

```

```

    for(;;){
        cntlvlka++;
        if(eca==0){
            break;
        }
    }

    time = (float) 0.0000000625*cntlvlka;
    jaraka = (float) (time*343)/2;
    jarakaCm = (float) (cntlvlka*31.133+26.98)/1000;
    levelka = (float) 50-jarakaCm;
    if(levelka<0){ levelka=0;}
}

void bacalevelki(){
    trb=1;
    delay_us(10);
    trb=0;

    for(;;){
        if(ecb==1){
            cntlvlki=0;
            break;
        }
    }

    for(;;){
        cntlvlki++;
        if(ecb==0){
            break;
        }
    }

    time = (float) 0.0000000625*cntlvlki;
    jaraki = (float) (time*343)/2;
    jarakiCm = (float) (cntlvlki*31.133+26.98)/1000;
    levelki = (float) 50-jarakiCm;
    if(levelki<0){ levelki=0;}
}

void setMov(unsigned int persen){
    data = persen*1.9+300;
    OCR1AH = (unsigned char)((((data & (unsigned int)0xFF00))>>8);

```

```

    OCR1AL = (unsigned char) (data & (unsigned int)0x00FF);
}

void setPmp(unsigned int persena){
    data2 = persena*2.5+250;
    OCR1BH = (unsigned char)(((data2 & (unsigned int)0xFF00)>>8);
    OCR1BL = (unsigned char) (data2 & (unsigned int)0x00FF);
}

void aktifmov(){
    DDRD.5=1;
    TCCR1A=0x82;
    TCCR1B=0x1B;
    TCNT1H=0x00;
    TCNT1L=0x00;
    ICR1H=0x03;
    ICR1L=0x02;
}

void nonaktifmov(){
    DDRD.5=0;
    TCCR1A=0x02;
    TCCR1B=0x1B;
    TCNT1H=0x00;
    TCNT1L=0x00;
    ICR1H=0x03;
    ICR1L=0x02;
}

void aktifpmp(){
    DDRD.4=1;
    TCCR1A=0x22;
    TCCR1B=0x1B;
    TCNT1H=0x00;
    TCNT1L=0x00;
    ICR1H=0x13;
    ICR1L=0x87;
}

void nonaktifpmp(){
    PORTD.4=1;
    TCCR1A=0x00;
    TCCR1B=0x00;

```

```

TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
}

```

```

void tesvalve(){
    lcd_clear();
    lcd_gotoxy(0,0);
    lcd_putsf("Tes Valve");
    for(;;){
        if(!up){
            lcd_gotoxy(0,1);
            lcd_putsf("Valve kiri ");
            delay_ms(300);
            vala=konek;
            valb=putus;
        }

        if(!dn){
            lcd_gotoxy(0,1);
            lcd_putsf("Valve kanan");
            delay_ms(300);
            vala=konek;
            valb=putus;
        }
    }

    lcd_clear();
}

```

```

void tesmov(){
    i=35;
    aktifmov();
    for(;;){
        if(!up){ i++; delay_ms(100);}
        if(!dn){ i--; delay_ms(100);}
        if(!ok){ delay_ms(200); lcd_clear(); break; }
        if(i>100) {i=10; delay_ms(10);}
        if(i<10) {i=100; delay_ms(10);}
        data = i*1.9+300;
        //OCR1AH = (unsigned char)(((data & (unsigned int)0xFF00)>>8);
        //OCR1AL = (unsigned char) (data & (unsigned int)0x00FF);
    }
}

```

```

        setMov(i);
        sprintf(buff,"MOV : %4d %%",i);
        lcd_gotoxy(0,0);
        lcd_puts(buff);
        sprintf(buff,"PWM : %4d ",data);
        lcd_gotoxy(0,1);
        lcd_puts(buff);
    }
    nonaktifmov();

}

void tespmp(){
    i=35;
    aktifpmp();
    for(;;){
        if(!up){ i++; delay_ms(100);}
        if(!dn){ i--; delay_ms(100);}
        if(!ok){ delay_ms(200); lcd_clear(); break; }
        if(i>100) {i=100; delay_ms(10);}
        if(i<10) {i=100; delay_ms(10);}
        data2 = i*2.5+250;
        //OCR1BH = (unsigned char)(((data & (unsigned int)0xFF00)>>8);
        //OCR1BL = (unsigned char) (data & (unsigned int)0x00FF);
        setPmp(i);
        sprintf(buff,"MTR : %4d %%",i);
        lcd_gotoxy(0,0);
        lcd_puts(buff);
        sprintf(buff,"PWM : %4d ",data2);
        lcd_gotoxy(0,1);
        lcd_puts(buff);
    }
    nonaktifpmp();

}

void menu(){
    awal:
    lcd_gotoxy(0,0); lcd_putsf("Main menu");
    lcd_gotoxy(2,1); lcd_putsf("Data EEPROM");
    lcd_gotoxy(2,2); lcd_putsf("Running");

```

```

menu1:
lcd_gotoxy(0,1); lcd_putchar(0x7E);
if(!dn){
    delay_ms(250); lcd_clear(); goto awal2;
}
if(!ok){
    delay_ms(250);
    lcd_gotoxy(0,1); lcd_putchar(0x20);
    goto menu2;
}
else goto menu1;

menu2:
lcd_gotoxy(0,2); lcd_putchar(0x7E);
if(!dn){
    delay_ms(250); lcd_clear(); goto awal3;
}
if(!ok){
    delay_ms(250);
    lcd_gotoxy(0,2); lcd_putchar(0x20);
    goto menu1;
}
else goto menu2;

awal2:
i=0;
lcd_gotoxy(0,0); lcd_putsf("EEPROM flow");
lcd_gotoxy(2,1); sprintf(buff,"Ki [%2d] :    L/s",i); lcd_puts(buff);
lcd_gotoxy(2,2); sprintf(buff,"Ka [%2d] :    L/s",i); lcd_puts(buff);
lcd_gotoxy(2,3); lcd_putsf("level");
lcd_gotoxy(11,1); ftoa(saveFlowKa[i],2,tempflowka); lcd_puts(tempflowka);
lcd_gotoxy(11,2); ftoa(saveFlowKi[i],2,tempflowki); lcd_puts(tempflowki);

lihat1:
lcd_gotoxy(0,1); lcd_putchar(0x7E);
if(!up){i++; delay_ms(200);}
if(!dn){i--; delay_ms(200);}
if(i>99) {i=0;}
if(i<0) {i=99;}
lcd_gotoxy(2,1); sprintf(buff,"Ki [%2d] :    L/s",i); lcd_puts(buff);
lcd_gotoxy(2,2); sprintf(buff,"Ka [%2d] :    L/s",i); lcd_puts(buff);
lcd_gotoxy(11,1); ftoa(saveFlowKa[i],2,tempflowka); lcd_puts(tempflowka);
lcd_gotoxy(11,2); ftoa(saveFlowKi[i],2,tempflowki); lcd_puts(tempflowki);

```

```

if(!ok){
    delay_ms(200);
    lcd_gotoxy(0,1); lcd_putchar(0x20);
    goto lihat2;}
else goto lihat1;

lihat2:
lcd_gotoxy(0,3); lcd_putchar(0x7E);
if(!dn){
    delay_ms(200);
    lcd_clear();
    lcd_gotoxy(0,3); lcd_putchar(0x20);
    goto levelawal2;}

if(!ok){
    delay_ms(200);
    lcd_gotoxy(0,3); lcd_putchar(0x20);
    goto lihat1;}
else goto lihat2;

levelawal2:
i=0;
lcd_gotoxy(0,0); lcd_putsf("EEPROM Level");
lcd_gotoxy(2,1); sprintf(buff,"Li [%2d] :   cm",i); lcd_puts(buff);
lcd_gotoxy(2,2); sprintf(buff,"La [%2d] :   cm",i); lcd_puts(buff);
lcd_gotoxy(2,3); lcd_putsf("back");
lcd_gotoxy(11,1); ftoa(saveLevelKa[i],2,templevelka); lcd_puts(templevelka);
lcd_gotoxy(11,2); ftoa(saveLevelKi[i],2,templevelki); lcd_puts(templevelki);

levellihat1:
lcd_gotoxy(0,1); lcd_putchar(0x7E);
if(!up){i++; delay_ms(200);}
if(!dn){i--; delay_ms(200);}
if(i>99) {i=0;}
if(i<0) {i=99;}
lcd_gotoxy(2,1); sprintf(buff,"Li [%2d] :   cm",i); lcd_puts(buff);
lcd_gotoxy(2,2); sprintf(buff,"La [%2d] :   cm",i); lcd_puts(buff);
lcd_gotoxy(11,1); ftoa(saveLevelKa[i],2,templevelka); lcd_puts(templevelka);
lcd_gotoxy(11,2); ftoa(saveLevelKi[i],2,templevelki); lcd_puts(templevelki);

if(!ok){
    delay_ms(200);
    lcd_gotoxy(0,1); lcd_putchar(0x20);

```

```

    goto levellihat2;}
else goto levellihat1;

levellihat2:
lcd_gotoxy(0,3); lcd_putchar(0x7E);
if(!dn){
    delay_ms(200);
    lcd_clear();
    lcd_gotoxy(0,3); lcd_putchar(0x20);
    goto awal;}

if(!ok){
    delay_ms(200);
    lcd_gotoxy(0,3); lcd_putchar(0x20);
    goto levellihat1;}
else goto levellihat2;

awal3:
lcd_gotoxy(0,0); lcd_putsf("Set mode :");
mode=0;

menumode:
if(!dn){ mode=0;}
if(!up){ mode=1;}
if(mode==0){ lcd_gotoxy(11,0); lcd_putsf("Lokal ");}
if(mode==1){ lcd_gotoxy(11,0); lcd_putsf("Android");}

if(!ok){
    delay_ms(200); goto exit1;
}
else goto menumode;

exit1:
lcd_gotoxy(0,1); lcd_putsf("Oke...");
delay_ms(500);
lcd_clear();
}

void setting(){
    lcd_gotoxy(0,0); lcd_putsf("Set Mov & Pmp");
    lcd_gotoxy(2,1); sprintf(buff,"Mov : %3d  %%",mov); lcd_puts(buff);
    lcd_gotoxy(2,2); sprintf(buff,"Pmp : %3d  %%",pmp); lcd_puts(buff);
    lcd_gotoxy(2,3); lcd_putsf("Oke");
}

```



```

menu1:
lcd_gotoxy(0,1); lcd_putchar(0x7E);
if(!up){ mov++; delay_ms(150); }
if(!dn){ mov--; delay_ms(150); }
if(mov>100){ mov=0; }
if(mov<30){ mov=100; }
lcd_gotoxy(2,1); sprintf(buff,"Mov : %3d  %%",mov); lcd_puts(buff);
saveMov=mov;
if(!ok){
    delay_ms(200);
    lcd_gotoxy(0,1); lcd_putchar(0x20);
    goto menu2;
}
goto menu1;

menu2:
lcd_gotoxy(0,2); lcd_putchar(0x7E);
if(!up){ pmp++; delay_ms(150); }
if(!dn){ pmp--; delay_ms(150); }
if(pmp>100){ pmp=0; }
if(pmp<30){ pmp=100; }
lcd_gotoxy(2,2); sprintf(buff,"Pmp : %3d  %%",pmp); lcd_puts(buff);
savePmp=pmp;
if(!ok){
    delay_ms(200);
    lcd_gotoxy(0,2); lcd_putchar(0x20);
    goto menu3;
}
goto menu2;

menu3:
lcd_gotoxy(0,3); lcd_putchar(0x7E);
if(!dn){
    lcd_gotoxy(6,3); lcd_putsf("Saving");
    delay_ms(200); lcd_clear(); goto exit1;
}
if(!ok){
    delay_ms(200);
    lcd_gotoxy(0,3); lcd_putchar(0x20);
    goto menu1;
}
goto menu3;

```

```

exit1:
lcd_gotoxy(0,0); lcd_putsf("Set Time & Level");
lcd_gotoxy(2,1); sprintf(buff,"Sve : %3d dt",periode); lcd_puts(buff);
lcd_gotoxy(2,2); sprintf(buff,"Spl : %3d cm",splvl); lcd_puts(buff);
lcd_gotoxy(2,3); lcd_putsf("Oke");
menu4:
lcd_gotoxy(0,1); lcd_putchar(0x7E);
if(!up){periode++; delay_ms(150);}
if(!dn){periode--; delay_ms(150);}
if(periode>360){periode=0;}
if(periode<0) {periode=360;}
lcd_gotoxy(2,1); sprintf(buff,"Sve : %3d dt",periode); lcd_puts(buff);
savePeriode=periode;
if(!ok){
    delay_ms(200);
    lcd_gotoxy(0,1); lcd_putchar(0x20);
    goto menu5;
}
goto menu4;

menu5:
lcd_gotoxy(0,2); lcd_putchar(0x7E);
if(!up){splvl++; delay_ms(150);}
if(!dn){splvl--; delay_ms(150);}
if(splvl>45) {splvl=0;}
if(splvl<0) {splvl=45;}
lcd_gotoxy(2,2); sprintf(buff,"Spl : %3d cm",splvl); lcd_puts(buff);
saveSplvl=splvl;
if(!ok){
    delay_ms(200);
    lcd_gotoxy(0,2); lcd_putchar(0x20);
    goto menu6;
}
goto menu5;

menu6:
lcd_gotoxy(0,3); lcd_putchar(0x7E);
if(!dn){
    lcd_gotoxy(6,3); lcd_putsf("Saving");
    delay_ms(200); lcd_clear(); goto exit2;
}
if(!ok){
    delay_ms(200);

```

```

    lcd_gotoxy(0,3); lcd_putchar(0x20);
    goto menu4;
}
goto menu6;

exit2:
lcd_clear();
}

void setwifi(){
    lcd_gotoxy(0,0) ; lcd_putsf("Set. via Android");
    for(;;){

    }
}

void kontrol(){
    if(levelki >= 34 || levelka>=34){
        buz=konek; delay_ms(500); buz=putus;
    }

    if(levelki < 14){
        indb=nyala;
    }

    if(levelki >= 20){
        indb=mati;
    }

    if(levelka < 14){
        inda=nyala;
    }

    if(levelka >= 20){
        inda=mati;
    }

    if(levelki>=splvl+1){
        valb=konek;
    }

    if(levelki<splvl-1){

```

```

    valb=putus;
}

if(levelka>=splvl+1){
    vala=konek;
}

if(levelka<splvl){
    vala=putus;
}

// Kontrol kanan
if(levelki > splvl+5){
    pmp=savePmp;
    pmp=10;
    aktifpmp();
    setPmp(10);delay_ms(500);
    nonaktifpmp();
}

if(levelki < splvl+5 && levelki >= splvl+2){
    pmp = savePmp;
    pmp=30;
    aktifpmp();
    setPmp(30);delay_ms(500);
    nonaktifpmp();
}

if(levelki < splvl+2 && levelki >= splvl-2){
    pmp = savePmp;
    aktifpmp();
    setPmp(pmp);delay_ms(500);
    nonaktifpmp();
}

if(levelki < splvl-2 && levelki >= splvl-5){
    pmp=70;
    aktifpmp();
    setPmp(70);delay_ms(500);
    nonaktifpmp();
}

```

```
if(levelki <= splvl-5){  
    pmp=90;  
    aktifpmp();  
    setPmp(90);  
    delay_ms(500);  
    nonaktifpmp();  
}
```

// Kontrol Kiri

```
    if(levelka > splvl+5){  
        mov=10;  
        aktifmov();  
        setMov(10); delay_ms(500);  
        nonaktifmov();  
    }
```

```
if(levelka < splvl+5 && levelka >= splvl+2){  
    mov=30;  
    aktifmov();  
    setMov(30);delay_ms(500);  
    nonaktifmov();  
}
```

```
if(levelka < splvl+2 && levelka >= splvl-2){  
    mov=saveMov;  
    aktifmov();  
    setMov(mov);delay_ms(500);  
    nonaktifmov();  
}
```

```
if(levelka < splvl-2 && levelka >= splvl-5){  
    mov=70;  
    aktifmov();  
    setMov(70);delay_ms(500);  
    nonaktifmov();  
}
```

```
if(levelka <= splvl-5){  
    mov=90;  
    aktifmov();  
    setMov(90); delay_ms(500);
```

```
    nonaktifmov();  
}  
}
```

```
void main(void)  
{
```

```
    setpin();
```

```
    TCCR0A=0x00;  
    TCCR0B=0x05;  
    TCNT0=0x64;  
    OCR0A=0x00;  
    OCR0B=0x00;
```

```
    TCCR1A=0xA2;  
    TCCR1B=0x1B;  
    TCNT1H=0x00;  
    TCNT1L=0x00;  
    ICR1H=0x13;  
    ICR1L=0x87;  
    OCR1AH=0x00;  
    OCR1AL=0x00;  
    OCR1BH=0x00;  
    OCR1BL=0x00;
```

```
    EICRA=0x0B;  
    EIMSK=0x03;  
    EIFR=0x03;  
    PCICR=0x00;
```

```
    TIMSK0=0x01;  
    TIMSK1=0x00;  
    TIMSK2=0x00;
```

```
    UCSR0A=0x00;  
    UCSR0B=0x08;  
    UCSR0C=0x06;  
    UBRR0H=0x00;  
    UBRR0L=0x67;
```

```
    pmp = savePmp;  
    mov = saveMov;
```

```

lcd_init(16);
lcd_putsf("Loading EEPROM... ");
for(i=0;i<20;i++){
    lcd_gotoxy(i,1);
    lcd_putchar(0xff);
    delay_ms(100);
}
lcd_clear();
if(!dn){delay_ms(350); tesmov();}
if(!ok){delay_ms(350); tesvalve();}
if(!up){delay_ms(350); tespmp();}

```

```

menu();

```

```

if(mode==0){setting();}
if(mode==1){setWifi();}

```

```

    lcd_gotoxy(0,0);
    lcd_putsf("Preparing MOV ");
    aktifmov();
    setMov(mov);
    for(i=0;i<20;i++){
        lcd_gotoxy(i,1);
        lcd_putchar(0xff);
        delay_ms(100);
    }
    nonaktifmov();
    lcd_clear();

```

```

    lcd_gotoxy(0,0);
    lcd_putsf("Preparing PMP ");
    aktifpmp();
    setPmp(pmp);
    for(i=0;i<20;i++){
        lcd_gotoxy(i,1);
        lcd_putchar(0xff);
        delay_ms(100);
    }
    nonaktifpmp();
    lcd_clear();

```

```

//input data history (Andiah A. 10511400000052)
moni=1;

```

```
lcd_gotoxy(0,0); lcd_putsf("Plant : ");  
printf("uphistory.php");  
delay_ms(200);  
msdetik=0; mili=0;  
#asm("sei")
```

```
while (1)  
{  
  
}  
}
```


Lampiran B

Listing Program Setting Wifi ESP8266

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <SoftwareSerial.h>

String inputString = "";
boolean stringComplete = false;
const byte rx=14;
const byte tx=12;
String input="";
const char* ssid    = "Prime7";
const char* password = "riha0895";
int httpCode ;
HTTPClient http;

WiFiClient client;
SoftwareSerial mySerial(rx, tx, false, 256);

void setup() {
  Serial.begin(9600);
  mySerial.begin(9600);
  WiFi.begin(ssid, password);
}

void loop() {
  while(mySerial.available()){
    char inChar = (char) mySerial.read();
    if(inChar=='\n'){
      inputString=inputString;
    }
    else if(inChar=='\r'){
      inputString=inputString;
    }
    else if(inChar=='\t'){
      inputString=inputString;
    }
    else{
```

```

        inputString += inChar;
    }

    if (inChar == '\n') {
        stringComplete = true;
    }

    if (stringComplete) {
        http.begin("http://myhyts.000webhostapp.com/"+inputString);
        Serial.println("http://myhyts.000webhostapp.com/"+inputString);
        httpCode = http.GET();
        http.end();
        //Serial.println(inputString);
        //mySerial.println(inputString);
        inputString = "";
        stringComplete = false;
    }
}

}

```

Lampiran C

Listing Program Aplikasi Android Studio

MainActivity.java

```
package app.hydrostatic.twl18;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Thread thread = new Thread() {
            public void run () {
                try{
                    sleep(3000);
                } catch (InterruptedException e){
                    e.printStackTrace();
                } finally {
                    startActivity(new
Intent(MainActivity.this, Login.class));
                    finish();
                }
            }
        };
        thread.start();
    }
}
```

Login.java

```
package app.hydrostatic.twl18;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class Login extends AppCompatActivity {
```

```

private Button login;
private Button exit;
private EditText txUser;
private EditText txPass;
String user, pass;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_login);
    getSupportActionBar().setTitle(" ");

    login = (Button) findViewById(R.id.btnLogin);
    exit = (Button) findViewById(R.id.btnExit);
    txUser = (EditText) findViewById(R.id.edUser);
    txPass = (EditText) findViewById(R.id.edPass);

    login.setOnClickListener(new View.OnClickListener()
    {
        @Override
        public void onClick(View view) {
            user = txUser.getText().toString();
            pass = txPass.getText().toString();
            if
            ((user.contains("tw118admin")) && ((pass.contains("TAlancarl23
            ")))) {

                Toast.makeText(getApplicationContext(), "Login
                Sukses", Toast.LENGTH_LONG).show();
                Intent intent = new Intent(Login.this,
                Home.class);

                finish();
                startActivity(intent);
            }
            else
            {

                Toast.makeText(getApplicationContext(), "Login
                Gagah", Toast.LENGTH_LONG).show();
            }
        }
    });
}
}

```

Home.java

```
package app.hydrostatic.tw118;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class Home extends AppCompatActivity {

    private Button database;
    private Button history;
    private Button keluar;
    private Button moni;
    private Button manual;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_home);
        getSupportActionBar().setTitle("HOME");

        database      = (Button) findViewById(R.id.btnData);
        history        = (Button)
findViewById(R.id.btnHistory);
        keluar         = (Button) findViewById(R.id.btnKeluar);
        moni           = (Button) findViewById(R.id.btnMoni);
        manual          = (Button)
findViewById(R.id.btnManual);

        database.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(Home.this,
Database.class);
                startActivity(intent);
            }
        });

        manual.setOnClickListener(new View.OnClickListener()
{
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(Home.this,
Manual.class);
```

```

        startActivity(intent);
    }
});

moni.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(Home.this,
Monitoring.class);
        startActivity(intent);
    }
});

history.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(Home.this,
History.class);
        startActivity(intent);
    }
});

keluar.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(Home.this,
History.class);
        moveTaskToBack(true);
    }
});
}
}

```

Database.java

```

package app.hydrostatic.twl18;

import android.content.Intent;
import android.net.Uri;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.webkit.WebView;
import android.widget.Button;

public class Database extends AppCompatActivity {

    private String urlHidro =

```

```

"https://myhyts.000webhostapp.com/tampilmoni.php";
    private String urlExport =
"https://myhyts.000webhostapp.com/export.php";
    private Button btnRefresh;
    private Button btnExport;
    private WebView vWeb;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_database);
        getSupportActionBar().setTitle("DATABASE");

        vWeb = (WebView)
findViewById(R.id.lihatWeb);
        btnRefresh = (Button)
findViewById(R.id.bthFresh);
        btnExport = (Button)
findViewById(R.id.btnExport);
        vWeb.loadUrl(urlHidro);

        btnRefresh.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                vWeb.loadUrl(urlHidro);
            }
        });

        btnExport.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new
Intent(Intent.ACTION_VIEW).setData(Uri.parse(urlExport)));
            }
        });
    }
}

```

Monitoring.java

```

package app.hydrostatic.tw118;

import android.os.Handler;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.webkit.WebView;

public class Monitoring extends AppCompatActivity {

```

```

        private String urlMoni =
"https://myhyts.000webhostapp.com/moni.php";
        private WebView vWeb;
        private Handler mHandler;
        @Override
        protected void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.activity_monitoring);
            getSupportActionBar().setTitle("MONITORING dan
SETTING");
            vWeb = (WebView)
findViewById(R.id.lihatWeb);
            vWeb.loadUrl(urlMoni);
            this.mHandler = new Handler();
            this.mHandler.postDelayed(m_Runnable, 5000);
        }

        private final Runnable m_Runnable = new Runnable()
        {
            public void run()
            {
                vWeb.loadUrl(urlMoni);
                Monitoring.this.mHandler.postDelayed(m_Runnable,
5000);
            }
        };
    }
}

```

History.java

```

package app.hydrostatic.twl18;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.webkit.WebView;
import android.widget.Button;

public class History extends AppCompatActivity {

    private String urlHistory =
"https://myhyts.000webhostapp.com/tampilhistory.php";
    private Button btnRefresh;
    private WebView vWeb;

    @Override
    protected void onCreate(Bundle savedInstanceState) {

```



```

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_history);
        getSupportActionBar().setTitle("HISTORY");
        vWeb = (WebView)
findViewById(R.id.lihatWeb);
        btnRefresh = (Button)
findViewById(R.id.bthFresh);
        vWeb.loadUrl(urlHistory);

        btnRefresh.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                vWeb.loadUrl(urlHistory);
            }
        });
    }
}

```

Halaman ini memang dikosongkan

Lampiran D

Datasheet Sensor Ultrasonik HC-SR04

ROBOT . HEAD to TOE

Product User's Manual – HC-SR04 Ultrasonic Sensor

1.0 INTRODUCTION

The HC-SR04 ultrasonic sensor uses sonar to determine distance to an object like bats or dolphins do. It offers excellent non-contact range detection with high accuracy and stable readings in an easy-to-use package. From 2cm to 400 cm or 1" to 13 feet. It operation is not affected by sunlight or black material like Sharp rangefinders are (although acoustically soft materials like cloth can be difficult to detect). It comes complete with ultrasonic transmitter and receiver module.

Features:

- Power Supply :+5V DC
- Quiescent Current : <2mA
- Working Current: 15mA
- Effectual Angle: <15°
- Ranging Distance : 2cm – 400 cm/1" - 13ft
- Resolution : 0.3 cm
- Measuring Angle: 30 degree
- Trigger Input Pulse width: 10uS
- Dimension: 45mm x 20mm x 15mm

ROBOT . HEAD to TOE

Product User's Manual – HC-SR04 Ultrasonic Sensor

4.0 PRODUCT SPECIFICATION AND LIMITATIONS

Parameter	Min	Typ.	Max	Unit
Operating Voltage	4.50	5.0	5.5	V
Quiescent Current	1.5	2	2.5	mA
Working Current	10	15	20	mA
Ultrasonic Frequency	-	40	-	kHz

3.0 PRODUCT LAYOUT

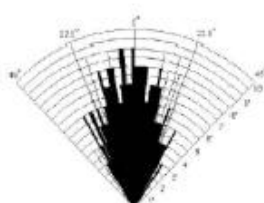
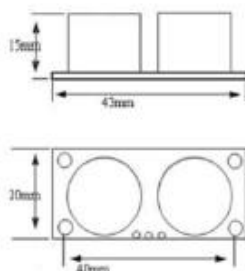


VCC = +5VDC

Trig = Trigger input of Sensor

Echo = Echo output of Sensor

GND = GND



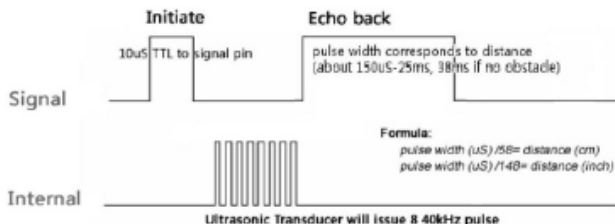
Practical test of performance,
Best in 30 degree angle

5.0 OPERATION

The timing diagram of HC-SR04 is shown. To start measurement, Trig of SR04 must receive a pulse of high (5V) for at least 10 μ s, this will initiate the sensor will transmit out 8 cycle of ultrasonic burst at 40kHz and wait for the reflected ultrasonic burst. When the sensor detected ultrasonic from receiver, it will set the Echo pin to high (5V) and delay for a period (width) which proportion to distance. To obtain the distance, measure the width (Ton) of Echo pin.

Time = Width of Echo pulse, in μ s (micro second)

- Distance in centimeters = Time / 58
- Distance in inches = Time / 148
- Or you can utilize the speed of sound, which is 340m/s



Note:

- Please connect the GND pin first before supplying power to VCC.
- Please make sure the surface of object to be detect should have at least 0.5 meter² better performance.

Halaman ini memang dikosongkan

Lampiran E

Datasheet Water Flow Sensor

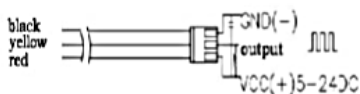


Specification

Mini. Working Voltage	DC 4.5V
Max. Working Current	15mA(DC 5V)
Working Voltage	5V~24V
Flow Rate Range	1~60L/min
Load Capacity	≤10mA(DC 5V)
Operating Temperature	≤80°C
Liquid Temperature	≤120°C
Operating Humidity	35%~90%RH
Water Pressure	≤1.75MPa
Storage Temperature	-25°C~+80°C
Storage Humidity	25%~95%RH

Wiring Diagram

The external diameter of thread the connections use is 1.4mm.



Output Table

Pulse frequency (Hz) in Horizontal Test= 4.8Q, Q is flow rate in L/min. (Results in +/- 3% range)

Output pulse high level	Signal voltage >4.5 V(input DC 5 V)
Output pulse low level	Signal voltage <0.5V(input DC 5V)
Precision	3% (Flow rate from 1L/min to 10L/min)
Output signal duty cycle	40%~60%

Halaman ini memang dikosongkan

Lampiran F

Datasheet Mikrokontroler AVR Atmega 1284P

Features

- High-performance, Low-power AVR[®] 8-bit Microcontroller
- Advanced RISC Architecture
 - 131 Powerful Instructions – Most Single-clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 20 MIPS Throughput at 20 MHz
 - On-chip 2-cycle Multiplier
- Nonvolatile Program and Data Memories
 - 128K Bytes of In-System Self-Programmable Flash
 - Endurance: 10,000 Write/Erase Cycles
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - 4K Bytes EEPROM
 - Endurance: 100,000 Write/Erase Cycles
 - 16K Bytes Internal SRAM
 - Programming Lock for Software Security
- JTAG (IEEE std. 1149.1 Compliant) Interface
 - Boundary-scan Capabilities According to the JTAG Standard
 - Extensive On-chip Debug Support
 - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
 - Two 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Six PWM Channels
 - 8-channel, 10-bit ADC
 - Differential mode with selectable gain at 1x, 10x or 200x
 - Byte-oriented Two-wire Serial Interface
 - Two Programmable Serial USART
 - Master/Slave SPI Serial Interface
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
 - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated RC Oscillator
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby
- I/O and Packages
 - 32 Programmable I/O Lines
 - 40-pin PDIP, 44-lead TQFP, and 44-pad QFN/MLF
- Operating Voltages
 - 1.8 - 5.5V for ATmega1284P
- Speed Grades
 - 0 - 4 MHz @ 1.8 - 5.5V
 - 0 - 10 MHz @ 2.7 - 5.5V
 - 0 - 20 MHz @ 4.5 - 5.5V
- Power Consumption at 1 MHz, 1.8V, 25°C
 - Active: 0.4 mA
 - Power-down Mode: 0.1 μ A
 - Power-save Mode: 0.7 μ A (Including 32 kHz RTC)



8-bit AVR[®]
Microcontroller
with 128K Bytes
In-System
Programmable
Flash

ATmega1284P

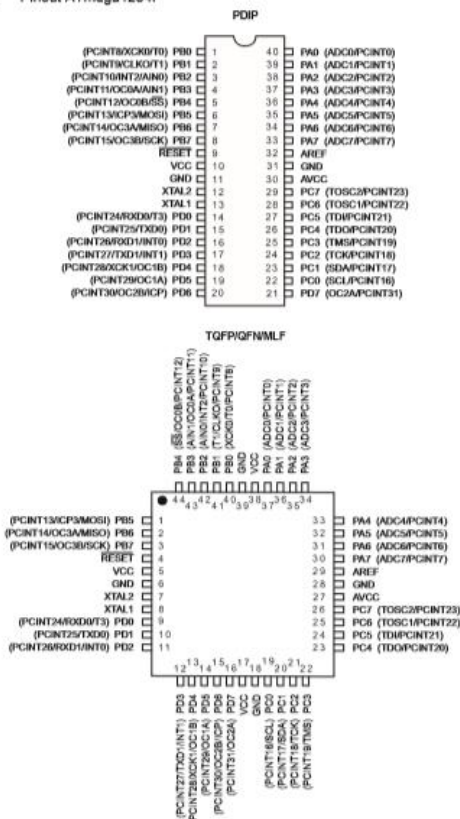
Preliminary

8059D-AVR-1 1/09



1. Pin Configurations

Figure 1-1. Pinout ATmega1284P



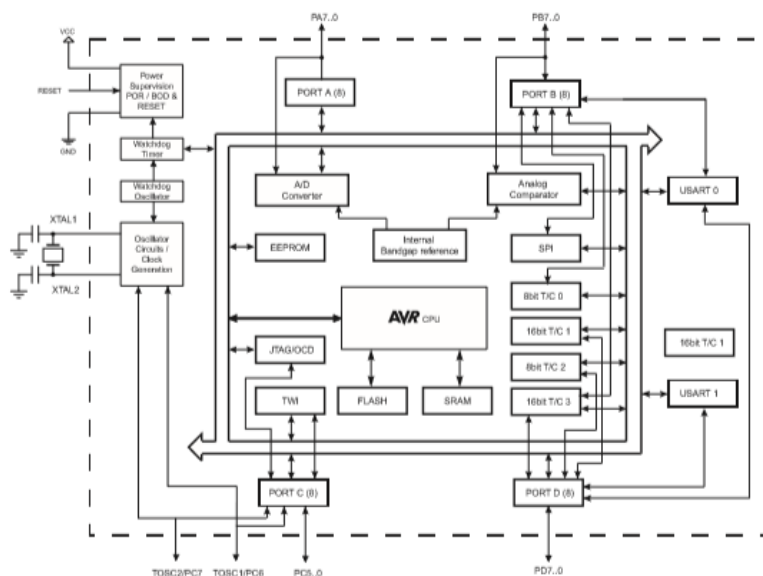
Note: The large center pad underneath the QFN/MLF package should be soldered to ground on the board to ensure good mechanical stability.

2. Overview

The ATmega1284P is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega1284P achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

2.1 Block Diagram

Figure 2-1. Block Diagram



The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega1284P provides the following features: 128K bytes of In-System Programmable Flash with Read-While-Write capabilities, 4K bytes EEPROM, 16K bytes SRAM, 32 general pur-

pose I/O lines, 32 general purpose working registers, Real Time Counter (RTC), three flexible Timer/Counters with compare modes and PWM, 2 USARTs, a byte oriented 2-wire Serial Interface, a 8-channel, 10-bit ADC with optional differential input stage with programmable gain, programmable Watchdog Timer with Internal Oscillator, an SPI serial port, IEEE std. 1149.1 compliant JTAG test interface, also used for accessing the On-chip Debug system and programming and six software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next interrupt or Hardware Reset. In Power-save mode, the asynchronous timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except Asynchronous Timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the Crystal/Resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low power consumption. In Extended Standby mode, both the main Oscillator and the Asynchronous Timer continue to run.

The device is manufactured using Atmel's high-density nonvolatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed in-system through an SPI serial interface, by a conventional nonvolatile memory programmer, or by an On-chip Boot program running on the AVR core. The boot program can use any interface to download the application program in the application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega1284P is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The ATmega1284P AVR is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, in-circuit emulators, and evaluation kits.

2.2 Pin Descriptions

2.2.1 VCC

Digital supply voltage.

2.2.2 GND

Ground.

2.2.3 Port A (PA7:PA0)

Port A serves as analog inputs to the Analog-to-digital Converter.

Port A also serves as an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port A pins that are externally pulled low will source current if the pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port A also serves the functions of various special features of the ATmega1284P as listed on [page 78](#).

2.2.4 Port B (PB7:PB0)

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port B also serves the functions of various special features of the ATmega1284P as listed on [page 80](#).

2.2.5 Port C (PC7:PC0)

Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port C also serves the functions of the JTAG interface, along with special features of the ATmega1284P as listed on [page 83](#).

2.2.6 Port D (PD7:PD0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port D also serves the functions of various special features of the ATmega1284P as listed on [page 86](#).

2.2.7 RESET

Reset input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in "System and Reset Characteristics" on [page 327](#). Shorter pulses are not guaranteed to generate a reset.

2.2.8 XTAL1

Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

2.2.9 XTAL2

Output from the inverting Oscillator amplifier.

2.2.10 AVCC

AVCC is the supply voltage pin for Port F and the Analog-to-digital Converter. It should be externally connected to V_{CC} , even if the ADC is not used. If the ADC is used, it should be connected to V_{CC} through a low-pass filter.

2.2.11 AREF

This is the analog reference pin for the Analog-to-digital Converter.

Halaman ini memang dikosongkan

Lampiran G

Datasheet Modul Wifi NODEMCU v3 Lua ESP8266



SYNACORP TRADING & SERVICES

No. 9, 1st Floor, Lrg 1/552, Bandar Tasik Mutiara, 14120 Simpang Ampat, S.Prai (S), Penang
Tel : +604.504.1617 Hunting Line : 012.4033.474 Fax : +604.502.1726
(Website) <http://www.synacorp.my> (Email) sales@synacorp.com.my

Arduino NODEMCU Lua IoT ESP8266 Wifi Controller Board v3



NodeMCU is a small board, based on the cheap ESP-12 Wi-Fi module containing a single-chip ESP8266 Wi-Fi SoC. The NodeMCU is an open-source firmware and development kit that helps you to prototype your IOT product within a few Lua script lines. The ESP-12 is one of the available modules containing an ESP8266 chip; it is becoming more and more popular because of its integrated Wi-Fi PCB antenna and shield.

- Open-source
- Interactive
- Programmable
- Low cost
- Simple
- Smart
- WI-FI enabled

FEATURES:

The Development Kit based on ESP8266, integrates GPIO, PWM, IIC, 1-Wire and ADC all in one board. Power your development in the fastest way combination with NodeMCU Firmware!

- USB-TTL included, plug & play
- 10 GPIO, every GPIO can be PWM, I2C, 1-wire
- FCC CERTIFIED WI-FI module (Coming soon)
- PCB antenna

